A HYBRID RATE CONTROL MECHANISM FOR FORWARDING AND CONGESTION CONTROL IN NAMED DATA NETWORK

MOHAMMED GAMAL ALSAMMAN

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2020

Awang Had Salleh
Graduate School
of Arts And Sciences

Universiti Utara Malaysia

PERAKUAN KERJA TESIS / DISERTASI
(Certification of thesis / dissertation)

Kami, yang bertandatangan, memperakukan bahawa
(We, the undersigned, certify that)

MOHAMMED GAMAL AHMED ALI ALSAMMAN

calon untuk Ijazah **PhD**
(candidate for the degree of)

telah mengemukakan tesis / disertasi yang bertajuk:
(has presented his/her thesis / dissertation of the following title):

"A HYBRID RATE CONTROL MECHANISM FOR FORWARDING AND CONGESTION CONTROL IN NAMED DATA NETWORK"

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.
(as it appears on the title page and front cover of the thesis / dissertation).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : ***05 Februari 2020.***
That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:
February 05, 2020.

Pengerusi Viva: *(Chairman for VIVA)*	**Assoc. Prof. Dr. Osman Ghazali**	Tandatangan *(Signature)*
Pemeriksa Luar: *(External Examiner)*	**Prof. Dr. Muhammad Shafie Abd Latif**	Tandatangan *(Signature)*
Pemeriksa Dalam: *(Internal Examiner)*	**Assoc. Prof. Dr. Mohd Hasbullah Omar**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Prof. Dr. Suhaidi Hassan**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Dr. Ahmad Suki Che Mohamed Arif**	Tandatangan *(Signature)*

Tarikh:
(Date) ***February 05, 2020***

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences

UUM College of Arts and Sciences

Universiti Utara Malaysia

06010 UUM Sintok

Abstrak

Rangkaian Data Bernama (NDN) adalah sebuah senibina Internet memuncul yang menggunakan senibina pengangkutan berasaskan-tarik, pengagregatan pada-laluan, hop-ke-hop, dan berbilang jalur. Oleh itu, algoritma-algorithma pengangkutan konvensional tidak akan berfungsi dengan betul dalam persekitaran NDN kerana lokasi sumber kandungan sering berubah. Perubahan ini meningkatkan cabaran kawalan penghantaran, dan ianya mempengaruhi penggunaan pautan, kesaksamaan, dan kestabilan rangkaian secara langsung. Kajian ini bertujuan untuk mencadangkan Mekanisme Kawalan Kadar Hibrid (HRCM) bagi mengawal kadar penghantaran dan kesesakan pautan untuk meningkatkan prestasi kebolehskalaan, kestabilan dan kesaksamaan rangkaian. HRCM terdiri daripada tiga skim, iaitu Pembentukan Defisit Pemberat Pusingan-Robin (SDWRR), Kelewatan Berbilang Jalur Selari (QPM) dan Kawalan Jelas Penyesuaian Tetingkap Konservatif berasaskan-Agil (EC-Agile). Skim SDWRR menjadualkan laluan berbeza pada antara muka penghala dengan mengesan dan memaklumkan kesesakan pautan. Skim QPM telah direkabentuk untuk menghantar paket Minat ke setiap laluan tersedia dengan menggunakan jalur lebar yang melahu. Skim EC-Agile mengawal kadar penghantaran dengan memeriksa setiap bingkisan yang diterima. Mekanisme HRCM yang dicadangkan ini telah dinilai melalui membandingkan dengan dua mekanisme yang berbeza, iaitu Kawalan Kesesakan Praktikal (PCON) dan Pembentukan Minat Hop-ke-hop (HIS) melalui simulasi ndnSIM. Penemuan menunjukkan bahawa HRCM meningkatkan kadar penghantaran dan kesaksamaan. HRCM mengatasi HIS dan PCON dari segi truput sebanyak 75%, masa lengah 20%, panjang giliran 55%, penggunaan pautan 41%, kesaksamaan 20%, dan masa muat turun 20%. Mekanisme HRCM yang dicadangkan menyumbang dalam penambahbaikan kadar penghantaran dan kesaksamaan NDN pada berbagai jenis aliran trafik. Oleh itu, skim SDWRR, QPM, dan EC-Agile ini boleh digunakan dalam pemantauan, pengawalan, dan pengurusan kesesakan serta penghantaran untuk Internet masa hadapan.

Kata kunci: Rangkaian data bernama, Strategi penghantaran, Pengurusan giliran, Kawalan kesesakan

Abstract

Named Data Networking (NDN) is an emerging Internet architecture that employs a pull-based, in-path caching, hop-by-hop, and multi-path transport architecture. Therefore, transport algorithms which use conventional paradigms would not work correctly in the NDN environment, since the content source location frequently changes. These changes raise forwarding and congestion control problems, and they directly affect the link utilization, fairness, and stability of the network. This study proposes a Hybrid Rate Control Mechanism (HRCM) to control the forwarding rate and link congestion to enhance network scalability, stability, and fairness performance. HRCM consists of three schemes namely Shaping Deficit Weight Round Robin (SDWRR), Queue-delay Parallel Multipath (QPM), and Explicit Control Agile-based conservative window adaptation (EC-Agile). The SDWRR scheme is scheduling different flows in router interfaces by fairly detecting and notifying the link congestion. The QPM scheme has been designed to forward Interest packets to all available paths that utilize idle bandwidths. The EC-Agile scheme controls forwarding rates by examining each packet received. The proposed HRCM was evaluated by comparing it with two different mechanisms, namely Practical Congestion Control (PCON) and Hop-by-hop Interest Shaping (HIS) through ndnSIM simulation. The findings show that HRCM enhances the forwarding rate and fairness. HRCM outperforms HIS and PCON in terms of throughput by 75%, delay 20%, queue length 55%, link utilization 41%, fairness 20%, and download time 20%. The proposed HRCM contributes to providing an enhanced forwarding rate and fairness in NDN with different types of traffic flow. Thus, the SDWRR, QPM, and EC-Agile schemes can be used in monitoring, controlling, and managing congestion and forwarding for the Internet of the future.

Keywords: Named Data Networks, Forwarding strategy, Queue management, Congestion control.

Acknowledgements

I will start by gratefully and sincerely thanking my supervisor Prof. Dr. Suhaidi Hassan for being my academic supervisor and allowing me to be a member of his research team. Also, I would like him to know that I appreciate all of his help, effort, encouragement and support; it was an honour and pleasure to work with him. I would also like to thank my co-supervisor Dr. Ahmad Suki Mohamed Arif for his invaluable contributions through this wonderful journey. He always found time to help me regardless of his own workload. I would also like to take this opportunity to thank all my brothers and sisters in the InterNetWorks Research Laboratory who strengthened me throughout my study. Special appreciation and thanks go to Dr. Alexander Afanasyev, who is a leader in the ndnSIM simulator as well as in NDN environments and Mr Klaus Schneider for there advice and help with all the problems I encountered during my work.

My gratitude to my father Gamal who has been my constant source of inspiration; and to my mother Nadaih who always has faith in me and prays for my success. To my wife Sarah and daughter Swan; thank you for your support, help and love. Thank you to my brother Ahmed and my sisters Eman and Sundus who are always willing to lend a helping hand. Also, thanks go to the Ministry of Yemen Higher Education and Yemen Telecommunication Ministry for their scholarship that gave me the opportunity to continue my study.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ACCPndn	-	Adaptive Congestion Control Protocol in Named Data Networking
ACCP	-	Adaptive Congestion Control Protocol
ACEF	-	Ant Colony Based Extensible Forwarding
ACKs	-	Acknowledgement
ACO	-	Ant Colony Optimization
AC-QoS-FS	-	Ant Colony based QoS-aware Forwarding Strategy
AIAD	-	Adaptive Increase Adaptive Decrease
AIMD	-	Additive Increase Multiplicative Decrease
AIM	-	Active Interest Management
AQM	-	Active Queue Management
ARM	-	Active Request Management
ARED	-	Adaptive Random Early Detection
AVQ	-	Adaptive Virtual Queue
BIF	-	Best Interface First
BIC	-	Binary Increase Congestion control
CCN	-	Content Centric Networks
CCS	-	Congestion Control Scheme
CCTCP		Content Centric Transport Control Protocol
CDNs	-	Content Delivery Networks
CHoPCoP	-	Chunk-switched Hop Pull Control Protocol
CMTB	-	Cooperative and Memory-efficient Token Bucket
CoDel	-	Controlling Queue Delay
CS	-	Content Store
cwnd	-	Congestion control window
DBN	-	Deep Belief Network

DONA	-	Data-Oriented Network Architectural
DRR	-	Deficit Round-Robin
DWRR	-	Deficit Weighted Round-Robin
EC-Agile	-	Explicit Congestion Agile-based
EC-CUBIC	-	Explicit Congestion control based CUBIC
ECN	-	Explicit Congestion Notification
ECP	-	Explicit Control Protocol
EIAIMD		Exponential Increase Addition Increase Multiplication Decline
EPF	-	Entropy-based Probabilistic Forwarding
FACE	-	Flow-Aware Congestion Estimation
FDCC	-	Fully-Distributed Congestion Control
FIB	-	Forwarding Information Base
FIFO	-	First-in-First-Out
FISP	-	Fair share Interest Shaping
FQ	-	Fair Queuing
FWT	-	Forwarding Table
GA	-	Genetic Algorithm
GCRBM	-	Gaussian–Bernoulli based Convolutional Deep Belief Network
H-ACK	-	Hop-by hop ACKnowledgment
HIS	-	Hop by hop Interest Shaping
HLAF	-	Heterogeneous-Latency Adaptive Forwarding
HoBHIS	-	Hop By Hop Interest Shaping
HRCM	-	Hybrid Rate Control Mechanism
HR-ICP	-	Hop by Hop-Interest Control Protocol
HWCC	-	Hop-by-hop Widow-based Congestion Control

ICNs	-	Information Centric Networks
ICP	-	Interest Control Protocol
IoT	-	Internet of Thing
IP	-	Internet Protocol
IS-IS	-	Intermediate System to Intermediate System
MADM	-	Multiple Attribute Decision Making
MANET	-	Mobile Ad-hoc NETwork
MFC	-	Multi-path Flow Control
MIAIMD	-	Multiple Increase Adaptive Increase Multiple Decrease
MIRCC	-	Multipath-aware ICN Rate-based Congestion Control
MLP	-	Multi Layer Perception
MPLS	-	Multi-Label Protocol Switching
MPTCP	-	Multipath TCP
MTU's	-	Maximum Transmission Unit's
NACK	-	Negative ACKnowledgement
NAT	-	Network Address Translation
NCFCC	-	Novel Cooperative and Fully-Distributed Congestion Control
NCE	-	Neighbor Cache Explore routing strategy
NCT	-	Neighbor Cache Table
NDN	-	Named Data Networks
NDO	-	Named Data Object
NetInf	-	Network of Information
NHBH-RCP	-	NDN Hop-By-Hop RCP
NSF	-	National Science Foundation
OMP-IF	-	On-demand Multi-Path Interest Forwarding
P2P	-	Peer-to-Peer
PAF	-	Probability-based Adaptive Forwarding

PARC	-	Palo Alto Research Center
PBTF	-	Probabilistic Binary Tree based Forwarding strategy
PCON	-	Practical CONgestion Control
PIT	-	Pending Internet Table
PI	-	Pending Internet
PMP-FS	-	Parallel Multi-Path Forwarding Strategy
PPT	-	Popularity Prediction Table
PQ	-	Priority Queuing
PrE	-	Probe Engine
PRTT	-	Predicted Round Trip Time
PSIRP	-	Publish-Subscribe Internet Routing Project
PSO	-	Particle Swarm Optimization
QoS-FS	-	QoS Forwarding Strategy
QPM	-	Queue-delay Parallel Multipath
RAAQM	-	Remote Adaptive Active Queue Management
RCP	-	Rate Control Protocol
RDPCC	-	Receiver Driven Performance based Congestion Control
RED	-	Random Early Detection
REM	-	Random Early Marking
RNN	-	Random Neural Network
RTO	-	Receiver Time Out
RTT	-	Round-Trip-Time
SAF	-	Stochastic Adaptive Forwarding
SAVQ	-	Stabilized Adaptive Virtual Queue
SDN	-	Software Defined Networks
SDWRR	-	Shaping Deficit Weight Round Robin.

SIRC	-	Self-regulating Interest Rate Control
SPP	-	Selection Probability Processor
SRED	-	Stabilized Random Early Drop
TARP	-	Traffic Aware Routing Protocol
TCP	-	Transport Control Protocol
TLFN	-	Time-lagged Feed forward Neural network
TSPDL	-	Time Series Prediction model based on Deep Learning
VIPs	-	Virtual Interest Packets
VRTT	-	Virtual Round Trip Time
WFQ	-	Weight Fair Queuing
WinCM	-	Window-based Congestion control Mechanism
WRR	-	Weighted Round-Robin
XCP	-	EXplicit Control Protocol

CHAPTER ONE
INTRODUCTION

Named Data Networking (NDN) [1, 2] is a new architecture for the future Internet that changes the technical protocols to assist applications with suggestions for economic, social and policy aspects of Internet ecosystem. Although NDN has undergone rapid growth in architectural design and application development, it is still very simple in terms of its architectural establishment. Active research is going on pivoted on the NDN project team supported by academia as well as industry. Considering the study of NDN is still in its beginning phase, several specific technologies are under study, such as content naming [3], routing [4, 5, 6], content caching [7] , forwarding [8, 9], congestion control [10], privacy protection and security [11]. In this research, the context of transport control refers to forwarding and congestion control as a whole to simplify the explanation as in [12, 13] and other researches.

Transport control is one of the most critical research areas in NDN today. Since NDN has no transport layer, the primary duty of the IP's transport layer has moved to the NDN forwarding plane. Also, NDN architecture properties as multi-path routing, in-network caching, new transmission modes like unit-cast, multicast and any-cast, increase the data transport complexity in future Internet. These properties have rendered the abundant literature on congestion control, multi-path forwarding and fairness of the IP architecture as no longer compatible.

This chapter is organized as follows: Section 1.1 provides the research background, while in Section 1.2 detailed descriptions of transport control architecture and research issues are presented. Section 1.3 presents the motivation of the study and the following section 1.4, highlights the problem statement of the research. Sections 1.5 and 1.6 cover the research questions and objectives respectively. The research scope is described in Section 1.7 and the significance and organization of the research in Section 1.8. Section 1.9 concludes the chapter.

1.1 Background

With continued growth in social networking, digital media, smart cities, smartphone applications and e-commerce, new expanded patterns of distribution networks are proposed for the Internet. Therefore, future Internet architecture design has new requirements and suggestions, such as mobility support, security, scalability, and reliability [14, 15]. Some solutions (e.g., Peer-to-Peer (P2P) and Content Delivery Network (CDN) systems) [16, 17, 18] have been introduced to satisfy part of these specifications with combined content-centric characteristics in the protocols as an overlay on the Internet Protocol (IP) network infrastructure. But, these information centric networks overlay are weak correspond to the Internet's communication-oriented underlay. This is because the IP network is based on the end-to-end communication model, which is not planned to maintain secure data distribution naturally, and it is inherently difficult to meet the condition of effective data distribution. Many of these critical challenges are pointedly rooted in initial underlying design arrangements in the Internet, and may not be resolvable without significant architectural alteration [19]. Most of the incremental functionality patches initially considering the obstacles were intended to be temporary solutions, that raised the level of complexity of the whole architecture [20, 21]. This incompatibility of today's Internet design motivates a new approach to developing the future Internet.

Several research projects have been developed to handle the future Internet designs and re-build the network architecture. Information-Centric Networking (ICN) is a new Internet approach that pursues the idea of decentralized networks. Moreover, ICN comes with a new structure that changes network addressing from the endpoint that uses IP to the concept of addressing the content itself, that is a content-based instead of the IP based approach. Several research communities have considerable interest in this subject, and various architectures have been widely studied [22]. Popular among these are Data-Oriented Network Architecture (DONA) [23], Content-Centric Networks (CCN) or Named Data Networks (NDN) [1], Publish-Subscribe Internet Routing Project (PSIRP) [24] and Network of Information (NetInf) [25]. Overall, these architectures are motivated by the current Internet challenges to re-examine, upgrade and maybe re-implement the current trends [22].

NDN is one of the projects that has gained massive interest in the ICN architectural research community which started its design and implementation. NDN reflects the general comprehension for qualities and restrictions of the current Internet architecture, which may perhaps be rolled out incrementally over the position of the existing Internet architecture. According to Jacobson et al. in [1], the hourglass architecture of the existing Internet centers on the universal network layer, named IP, where the minimum functionality necessary is implemented for global interconnectivity. The hourglass has a thin waist that accommodates and amplifies the Internet's explosive growth, in such a way that the upper layer and lower layer technologies are independently allowed to develop without unnecessary constraints. Although NDN retains the same hourglass shape architecturally as in current Internet architecture, the thin waist of hourglass is it changes to concentrate specifically on content rather than location. Moreover, the semantics of the network communication is also changed from delivering or retrieving a packet to a specific destination address to delivering or retrieving a content that is identified by a name.

To achieve this, the two types of packets are maintained in NDN communication architecture, the Interest packet and the Data packet. The first type which is Interest packets are generated by the consumer to retrieve content from the producer, sent fragmented as Data packets which is the second type. The NDN router facilitates its functionality by maintaining three noteworthy data structures: tables, such as Content Store (CS), a Pending Interest Table (PIT), and a Forwarding Information Base (FIB). The FIB table is contained name-prefix according to the routing protocol. The name-prefixes are used to guide the Interests packets toward content providers or publishers. The PIT contained and cached of all Interests prefixes which have been sent but have not yet been fulfilled. Hence, when an NDN router gets the same Interest packets from a downstream subscriber's nodes, it considered the first one and forward to the upstream towards the data publisher. The CS is a temporary storage that caches and store Data packets that have received by the NDN routers. Since the Data packet in NDN architecture is meaningful despite being independent of its origin or destination, it can be stored to fulfil future Interests. Chapter two offers more detail about NDN, involving several changes to router architecture and functionality [1].

1.2 Transport control in NDN

Since the required message for transport is embedded inside the data names [26], the sequence numbers and ports are absence in NDN, also not appears to own an independent transport layer. Instead, NDN possess two-fold of transport control such as fundamental forwarding strategy and transport functionality, which can be considered as flow control, congestion control and scheduling. The transport control in NDN is performed by combining libraries of applications on the user side and the strategy module on the router side. The transport control provides resilient and reliable delivery at the application layer by monitors and sustains the status of the incoming Data packet (e.g., a given time threshold). If the consumer still wants the unsatisfied Interest packets, it re-transmits them after some time threshold. On the router side, the transport control monitor controls the status of the transmission queue and forward adaptation [27, 12]. Both sides are discussed briefly in the following sub-sections.

1.2.1 Consumer control

Since NDN serves as consumer-driven "Pull" mode, where the consumer demands content utilizes the Interest packets by forwarding it to the content producer nodes. Immediately the producer returns the requested Data packets back to the consumer as a satisfaction of the request demanded. The simplest way to control congestion that happen by the Data packet flow is by controlling the consumer's Interest packets rate, as in TCP that used window-based Interest packets transference. However, contrary to TCP, a single type of window is required in NDN, that serves the function of both notification window and congestion window. That is the number of Data packets permitted to be sent and the number of Interest packets allowed to be sent, due to the fact that NDN can control the Data packet flow at the consumer. For regulating the window size, the Adaptive Increase Multiple Decrease (AIMD) mechanism may be used. when a full window of Interest packets is acknowledged by Data packet response, the Congestion Control Window (cwnd) is increased, and decreased when a Round Time Out (RTO) timeout occurs [12, 10].

Considering NDN packets are named individually in such a way that each Interest packet corresponds to one Data packet, an organized packet transport is not needed. Also, for every demand NDN employs positive acknowledgement independently instead of cumulative acknowledgements (ACKs) of TCP. The sender or provider continues to send successive Data packets matching the Interests it has received without duplicates of the Data packets, even if there is an Interest packet loss like TCP's duplicated ACKs. Instead, NDN multi-point distribution may cause duplication. Consequently, by duplicated Data packets the receiver cannot detect any congestion [12, 10].

For that, the detailed cwnd adjustment mechanism needs to be studied as the duplicate ACKs for TCP's Quick Recovery mechanisms are not applicable in NDN. In addition, the NDN network supports multi-homing, that the network can have multiple source for one content which may be served from in-network caches. Hence, by using a single RTO timer the receiver is incapable to detect congestion. However, the initial studies failed to consider these matters. Accordingly, the existing control mechanisms was employed to whether recognized multi-source based on the receiver, which is divided into two mechanism called single-source and multi-source.

1.2.2 Router side control

NDN architecture has a manageable forwarding plane, which allows researchers to investigate several NDN routers. Forwarding strategy used to manage the flow load by restraining PIT size, NDN's built-in caching and the individual hop-by-hop flow balance ; these methods can efficiently avoid and alleviate network congestion control. In addition, the transport control used the built-in multicast in FIB increases for the improvements of the efficiency and flexibility. So, combining backpressure mechanisms with hop-by-hop Interest packets shaping shows to be more viable alternative for congestion control in NDN than the usual TCP mechanisms. Purposely the hop-by-hop congestion control is employed for bypassing congestion proactively and react adequately when congestion occurs. The next parts describe the design of hop-by-hop transport control individually from the aspects of routing and forwarding, queue management and Interests scheduling mechanisms.

Interests scheduling: The focus on the Interest packet shaping at each router is the most critical and primary building block of the Interest packet shaping used in the network. The concept is straightforward with the aim of bypassing the congestion in the downstream link, which makes each router shapes the Interest packets and sends them to upstream links to constrain the returning Data packet rate to achieve the aim. Each router manages a rate limit for each uplink interface for each name prefix. The network can allocate different priorities and implement different services for each prefix. However, it also places more load on the node, which previously involved many tasks like PIT and FIB records management, signing the cache policy and table's looking up.

Further, this obstacle is confused by certain circumstances like traffic burst, varying packet size, asymmetric link bandwidth and different return time Data packets. Hence, a more careful and intensive review for this unusual relationship between Interest and Data packets is required. There are many mechanisms, discussed in Chapter 2, that are delayed the Interest packets for ensuring the congestion avoidance or decision. The specific adjustment mechanism for setting the Interest packets' sending limit requires further study.

Queue management: This method is applied in the NDN router to determine when and which packets are to be dropped at the output interface when congestion is anticipated. Also, in NDN transport control the AQM for IP networks can be implement with some modification to achieve better performance in NDN transport control. These modifications are as follows. Firstly, build two type of queue one for Interest and the other for Data packet. With correct Interest packet shaping, the downlink will not be overloaded by Data packets. Thus, Interest packets shaping transfers the Data queue indication into an Interest queue congestion indication. Second, it shows that an explicit congestion signalling mechanism is a more reliable way to help the receiver with the Interest rate adjustment. However, this kind of mechanism typically experiences the problems of congestion signals themselves. As congestion signalling may aggravate congestion or become lost as a result of the congestion, NDN employs

AQM mechanisms as state-of-the-art to generate NACKs promptly, and alleviate the congestion signalling mechanism for not suffering from the traditional problem. With NACK, such problems are automatically solved as the packet is sent to the customer precisely as like the returning Data packet. Also considered that when a neighbouring router sends an Interest packet, sufficient bandwidth is required to send the returning Data packet in the opposite direction of the link. As long as NACK should never become lost through congestion because it is smaller than Data packets.

Routing and forwarding mechanisms: The routing plan in NDN are the component that decides on the available routes for the forwarding plane [28]. The link status and distance-vector of the standard routing mechanisms can be adjusted to NDN routers. NDN router publishes name prefixes instead of publishing IP prefixes and manage names as a series of opaque elements. The NDN routing protocol creates these messages as routing prefixes across the network and inform each router to manage it as FIB [26]. The NDN-based link-state routing (NLSR) is the first routing protocol designed for NDN [29]. The routing protocol of NDN can employ any underlying transmission channel such as TCP/UDP tunnels, IP tunnels, and Ethernet to interact with routing messages. Routing preference is updated when routers forward Interest packets in which the update information is carried to recover Data packets. Most importantly, NLSR formulates in each router a name-based FIB to maintain NDN's forwarding plane.

By contrast, the most critical role in NDN routers is performed by forwarding strategy module because of its efficiency and resilience. The role is performed by joining the elements of PIT and FIB, for the execution of the adaptive forwarding mechanisms that is designed to bypass congestion and failures. Moreover, to determine the network conditions, the central multipath forwarding performs congestion control and load balancing. Hence, the efficient way to explore the superior capabilities was proposed initially in [30], in the study, the FIB entries and information stored in PIT are explained based on how to manage this information to produce forwarding arrangements based on network requirements. Thus, the flexibility and robustness of forwarding plane is one of the essential advantages of NDN transport control.

1.3 Motivation

As explained at the start of this chapter, although NDN has undergone rapid growth in architectural design and application development it is still immature architecturally. Active research concentrating on the NDN project team is supported by both academia and industry, with specific issues are still under consideration including content naming [3], routing [4, 5, 6], transporting, content caching [7] and forwarding [8, 9], security and privacy protection [11].

In transporting issues, as the primary duty of the IP's transport layer has moved to the NDN forwarding plane, since NDN has no transport layer. Also, the new properties as multi-path routing, in-network caching replication, transmission modes like unicast, multicast and any-cast, increase the data transport complexity in the Internet. These properties have changed the numerous literature on congestion control, multipath forwarding and fairness of the current architecture as no longer compatible.

A recent study [8, 31, 32, 33, 34, 35, 36, 37] proposed using multiple RTT estimators at the receivers to gauge network congestion of each path, to deal with challenges caused by the multiple-source and multiple-path transfer. Other studies [38, 39, 40, 41, 13, 42, 43] suggest a hop-by-hop Interest shaping scheme to prevent network congestion actively. Despite many techniques proposed to investigate the functionalities of NDN, there is a need to explore the functions of flow control, congestion control, and how to make use of the multi-path ability in a better way. Therefore, there are many possibilities for designing a transport control to suit different network conditions and problems, such as wired environments versus wireless environments. Exploring different alternatives for transport control is an active research field, including

i. Selecting suitable forwarding control that is aware of different packet sizes and bandwidths.
ii. Selecting appropriate indicators to prevent link congestion.
iii. Finding appropriate metrics to rank interfaces.
iv. Adapting indicators to balance traffic load and congestion.
v. Avoiding instability while sustaining best data delivery performance.

Since the transport control module at each node represents a vital role in efficient data retrieval and reducing inefficient probing on different links, the research in NDN transport control deserves to be studied further.

1.4 Problem Statement

In NDN, Interest packets aggregation minimizes the network load when several Interest packets from different sources request the same Data packets, by forwarding a single Interest packet upstream and forwarding multiple Data packet to downstream. However, if the NDN network is unable to control the forwarding rate as current networks do in TCP, in a congested network this leads to a deterioration in network scalability. Multiple congestion control mechanisms have been proposed to control the forwarding rate of Interest packets and choose appropriate interfaces to which to forward them without congestion [12, 10].

Some of these mechanisms apply the delay base Round Trip Time (RTT) control mechanism in the consumer application to control the rate [8, 31, 32, 33, 34, 35, 36, 37, 44]. As NDN natively supports multi-source and multi-path communications that result from multi-RTT timers, these delay-based mechanisms have difficulty in detecting the congestion as they use a single RTT. Nevertheless, delay-based congestion control mechanisms are inherent in the late loss recovery problem because there is no way to discard PIT entry in NDN routers when a Data packet is lost. However, waiting for the expiry of the Interest lifetime to resend the Interest packet is not appropriate for delay-sensitive applications that require high throughput and minimum delay [30]. Furthermore, the consumer control based is not sufficient to detect available bandwidth in the network and ensure fairness between other flows [10, 12, 45].

Other mechanisms are rate-based intermediate routers to detect congestion and make an explicit notification to the consumer by marking the Data packets or generating NACK packets [30, 46, 47, 48, 49, 13, 50, 51, 52, 53]. However, these types of mechanism rely on the consumer side, only monitoring congestion in the router without taking any action to delay the Interest in the router. Furthermore, the mechanisms that marking Data packets did not consider the Interest aggregations. When the Data packet is received by the NDN router, it is forwarded to all downlink routers who request it, and when a router receives the marked Data, it reduces its rate even if some routers are not congested. This affects network stability and links utilization.

Nevertheless, other mechanisms control the congestion in NDN using another type of hop-by-hop control called Interest shaping control [38, 39, 40, 41, 13, 42, 43]. However, these shaping mechanisms are not independent as they need to know the bandwidth, queue size and the delay of the links in advance to control the forwarding rate, and these parameters are not applicable in real networks. Furthermore, all shaping mechanisms use queue length to indicate congestion without considering the variation of queue size in routers [54, 51]. Delay-sensitive applications are affected by long queue size. Some mechanisms [51, 55] used Controlling Queue Delay (CoDel) to overcome the queue size variation and buffer bloat. Although, in terms of fairness this needs to be improved by combining the CoDel algorithm with a scheduling algorithm such as Fair Queue mechanism [56, 57, 58].

Most of these mechanisms are forward based on the best path of each forwarding mechanism ranking procedures, but other mechanisms adapt the multipath forwarding strategy to control congestion by distributing the Interest packets to other paths. Some mechanisms [59, 60] rank interfaces based on the delay in arrival of Data packets but as these are based on RTT they face the problems of variation of RTT as mentioned. Other mechanisms [35, 61] with ranked interfaces based on time-out have the same problem of content multi-homing. In addition, [62, 63, 64] proposed mechanism that depends on the number of pending Interest packets at each available interface to distribute the load. Although pending Interest is a local parameter and does not affect content multi-homing and variation delay, the forwarding ratio will be affected by multiple Interest flow and unknown network behaviour. Other mechanisms [51, 55, 65] rank each path to overcome the problem of multi-home content and network behaviour, but waiting for feedback leads to a sub-optimal forwarding ratio because the feedback notification may be delayed or dropped.

In conclusion, because of in-path caching RTT as an indicator would affect the forwarding rate in return would affect the utilization of available links [41, 66, 51, 55]. Also, because of the different size of queues and packets in the network using inappropriate scheduling and indicator would affect the forwarding rate and fairness [56, 57, 58]. Especially, with the continuous growth of delay-sensitive applications on the Internet like digital media, Interest of things and vehicle to a vehicle that requires high throughput and minimum delay. Also, using one explicit feedback will not be enough to overcome the packet aggregation. Nevertheless, the time variation between incoming packets and the massive number of them would increase the consumer rate fluctuation that would affect the forwarding stability [45, 41, 38, 13]. Therefore, this research needs to propose a mechanism to overcome the multi-homing content, variation of queue size, packet aggregation and severe fluctuations while ensuring the link utilization.

1.5 Research Question

To implement a stable transport control mechanism is required to serve as a smart load distributor within a network domain in a continuous manner and offload congested links and paths. These raise the following questions:

i. How to avoid link congestion while utilizing the available bandwidth and ensure fairness between flows in the Named Data Networking environment?

ii. What techniques are required to implement a stable forwarding and congestion mechanism in the Named Data Network?

iii. Which method can suitably be used to implement and evaluate the proposed mechanism?

1.6 Research Objectives

This research aims to design a new rate control mechanism that distributes the load within a network domain in a continuous manner and offloads congested links and paths. This mechanism should be able to improve the forwarding mechanism by minimizing the link and caching overheads, because it forwards the Interest in multipath, and avoiding congestion. This mechanism also considers fairness among different types of flow. This research therefore has the following objectives:

i. To propose a hop-by-hop shaping scheme with explicit congestion feedback functions to avoid link congestion and ensure flow fairness in NDN environments.
ii. To propose parallel multipath forwarding scheme to utilize available links in NDN environments.
iii. To propose a consumer rate adaptation scheme to improve forwarding stability in NDN environments.
iv. To implement by integrating the proposed schemes for improving link utilization, fairness and stability in NDN environments.

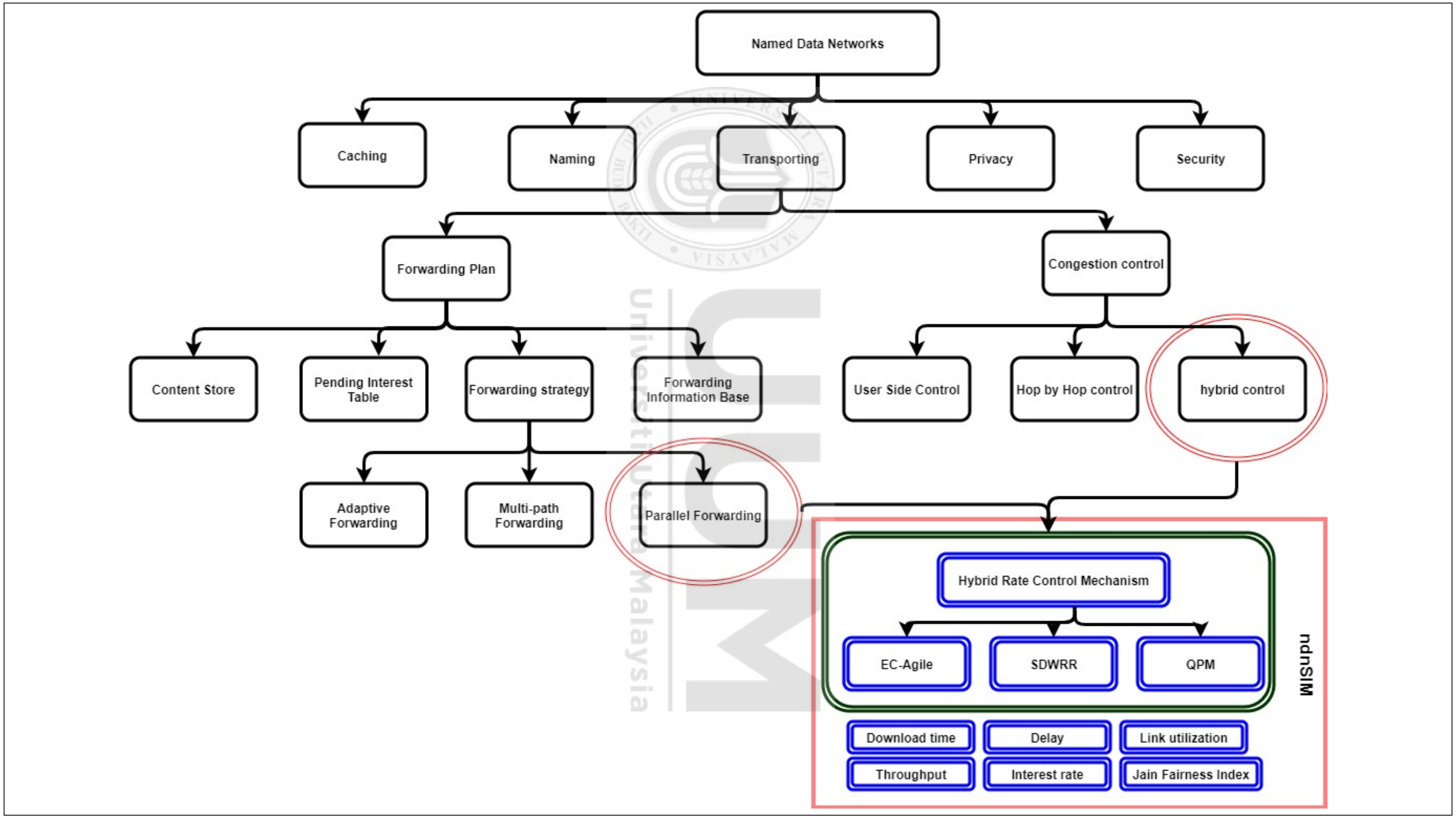


Figure 1.1. Research Scope

1.7 Research Scope

The entirety of this study is focused on transport control in Named Data Networks, as demonstrated in Figure 1.1. In the forwarding part, this research focuses on using a parallel forwarding technique to forward packets to up-links. While a hybridize the consumer and in-path router to control the rate and prevent congestion. Therefore, this research has three components that can affect the transport control performance based on the NDN architecture: Shaping Deficit Weight Round Robin (SDWRR) to control the rate in the router side, Queue-delay Parallel Multipath (QPM) to distribute incoming rate to all available path and Explicit Congestion Agile-based Conservative Window Adaptation (EC-Agile) to control the rate in consumer side. The research evaluates several performance metrics of the proposed mechanism for NDN architecture using a ndnSIM simulator: link utilization, throughput, download time, delay, Interest rate, and Jain fairness index. However, other NDN issues, such as caching, naming, privacy and security, are outside the scope of this research.

1.8 Significance of the Research

This research introduces a HRCM comprising SDWRR, EC-Agile and Queue-delay Parallel Multipath forwarding (QPM) schemes. The proposed schemes are capable of monitoring the queues between hop routers to control the forwarding interest and prevent congestion in the downlink and uplink, providing fairness and stability. Finally, HRCM will be able to provide a high-link utilization with multiple flows fairly and stably with low packet loss.

1.9 Organization of the Thesis

Chapter One: An overview of NDN and the thesis as a whole. It covers the motivation, problem statement, objective and scope of the research.

Chapter Two: The literature review of previous and current related work. It presents relevant information for better understanding of the selected research area.

Chapter Three: Explains the methodology and research design used to achieve the research objectives. It presents combinations of several schemes in order to propose and implement HRCM. The experimental design, verification, validation, and evaluation are outlined.

Chapter Four: Elaborates the design of the proposed mechanism. It describes the full deployment and techniques for achieving the HRCM components (SDWRR, EC-Agile and QPM), including the verification, validation and evaluation of each, and justifying their inclusion.

Chapter Five: Develops HRCM and presents in detail a comprehensive evaluation of the proposed mechanism through simulations. It examines the HRCM with different simulation scenarios for different network typologies. Theoretical and graphical comparisons of HRCM with current solutions are made.

Chapter Six: States the conclusion and reviews how the primary research goals have been achieved. It highlights the main contributions of the thesis, as well as possible directions for future work based on the findings of the current study.

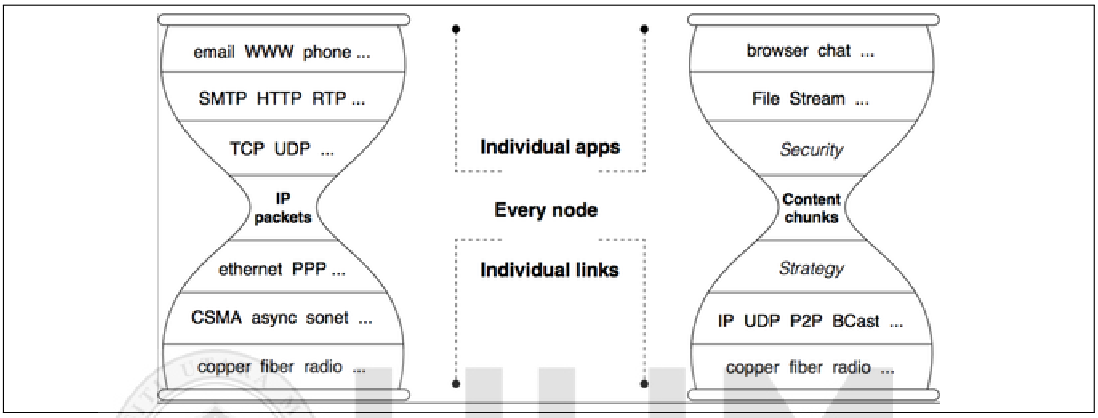
CHAPTER TWO
LITERATURE REVIEW

Work on a proposed new architectural design for the future Internet, termed Named Data Networking (NDN), is a joint research effort. Since the NDN research is at an early stage, a number of specific functions are still under study, including content naming [3], routing [4, 5, 6], transporting, content caching [7] and forwarding [8, 9], security and privacy protection [11]. Since NDN has no transport layer, the primary duty of the IP's transport layer has moved to the NDN forwarding plane. Also, NDN architecture properties such as multi-path routing, in-network caching, replication, new transmission modes like uni-cast, multicast and any-cast, increase the data transport complexity in future Internet. These properties have rendered the abundant literature on congestion control, multi-path forwarding and fairness of the IP architecture as no longer compatible [12, 64].

The research plan was clarified in Chapter One, whereas this chapter describes in more significant detail the background and several important research-related issues of transport control design, implementation, and management, which assist in defining the general framework of this research. The chapter also focuses on analyzing the NDN communication system, particularly the architecture, features and application. In-depth analysis and a critical review of transport control implementation includes user-side control and hop-by-hop shaping. Accordingly, the chapter is organized as follows. Section 2.1 is a detailed description of NDN architecture, and the transport control concept is introduced in Section 2.2. Section 2.3 describes in detail transport control techniques that fall within the research scope. Existed techniques are critically reviewed in Section 2.4, and theories pertinent to the proposed mechanism are described in Section 2.5. The chapter ends with a summary in Section 2.6.

2.1 NDN Architecture

NDN architecture modifies the Internet Protocol (IP) layer hourglass by presenting the layer with content names, as shown in Figure 2.1; other layers are unchanged. Due to the architectural modification of the IP layer, the routing principle is changed from the end-to-end point (which depends on the IP) to the consumer principle that depends on named data.



TCP/IP NDN

Figure 2.1. Difference between TCP/IP and NDN [1]

This supports the retrieval of data from the producer to the consumer (and vice versa), gaining the advantages of multipath direction and loop-free transmission. This architecture means that the consumer does not necessarily need to specify the location from which the contents are coming [2]. To carefully analyze the NDN architectural design, some important functional characteristics and important auxiliary support are presented in this section as a proposed taxonomy of NDN.

2.1.1 Packets Types in NDN

NDN architecture introduces two types of packet, as shown in Figure 2.2. The first is the Interest packet; by sending these over the outgoing interfaces, the NDN router announces the subscriber node's demand for the content that is named by the Interest packet. The Interest packet is broadcast to the available interfaces in order to retrieve the Data packet, which usually contains the desired content name by default. The Interest packet is accompanied by selective information, such as the scope inside the network and from where the Data packet has to come. The Interest packet also has a nonce used for detecting duplicate Interest packets.

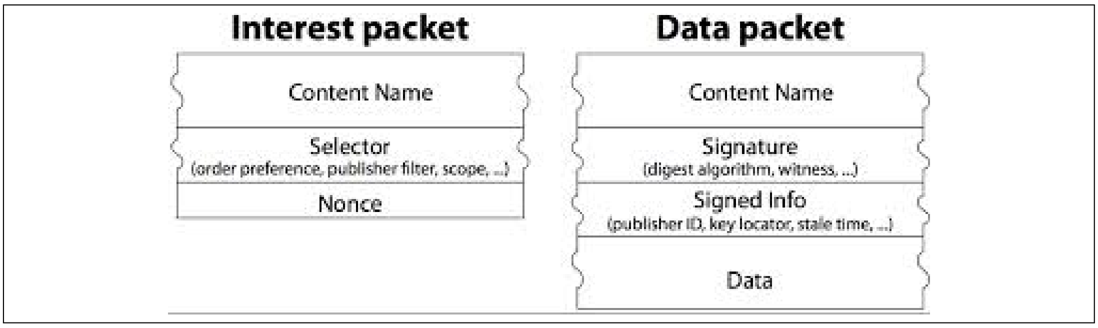


Figure 2.2. Named Data Network Packets Format [1]

The second packet, the Data packet, is used in response to an incoming Interest packet. Data packets are required to satisfy an Interest packet in that they have a one-to-one relation, where the Interest packet is consumed by the Data packet. This rule of thumb keeps a flow balance at each hop and blocks congestion in the middle of a connection path. Content names in NDN are structured hierarchically. This Data packet serves one Interest packet if its name prefix and the name of the Interest packet match. Apart from the arbitrary binary data and name, the Data packet also has a digital signature with a certain cryptographic digest to signed information. This last field provides extra information about the packet, including the publisher's ID to locate the key for checking the time stamp or signature. With these ways of verification, it is guaranteed that the packet identifies and authenticates itself, and there is no need of legitimacy from the channel by which it is transferred [1, 2].

2.1.2 Tables Types in NDN

Each NDN router uses three data structures in processing packet forwarding: the Content Store, Forwarding Information Base and Pending Interest Table [1, 2] as shown in Figure 2.3.

- Content Store (CS) is a cache structure (buffer memory) in a NDN router that stores chunks for a very long time would be updated by applying cache policies. As content is self-authenticating as well as self-identifying, each one of the packets should be useful to potential participants nearby in the network. The ability to serve content directly rather than generating additional lookups reduces total bandwidth usage as well as latency.
- Forward Information Base (FIB) is used for storing information on packet forward. It is like a routing table in the common IP router. FIB stores information on which interfaces Interest packets are forwarded upstream towards the source of the content. Hence, the design enables more than one entry that may be needed to be queried in parallel since forwarding is not limited to one spanning tree.
- Pending Interest Table (PIT) consists of the arrival interfaces of Interest packets which have been forwarded, but which are still waiting for a matching Data packet. This information is needed in order to deliver Data packets to their subscribers. To increase the PIT utilization, the entries need to be timed out quickly within the Interest packet lifetime.

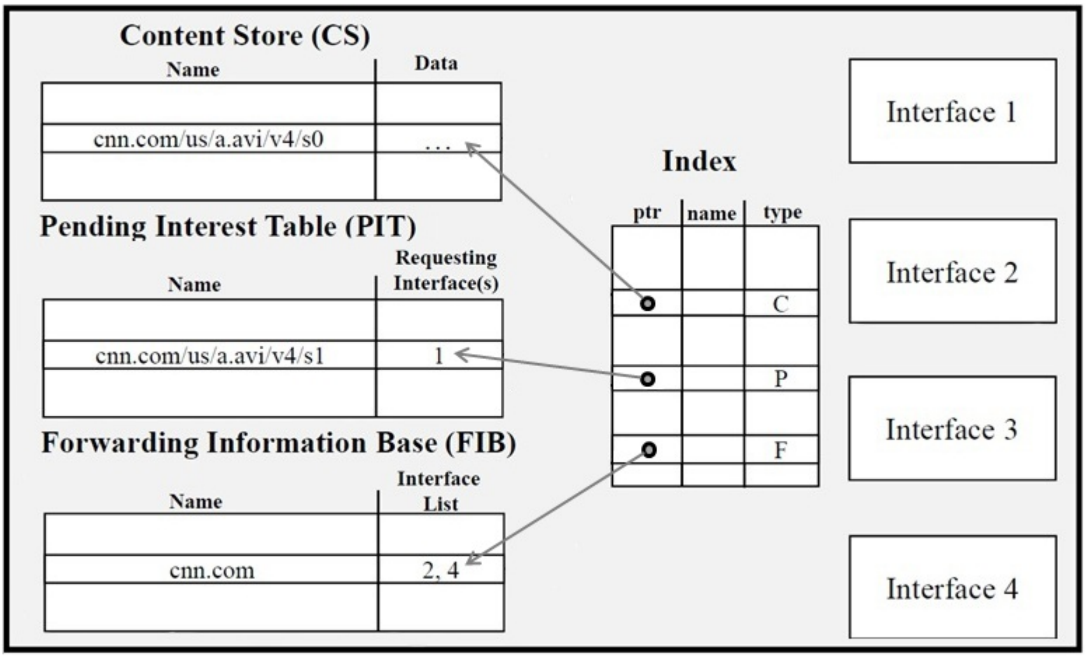


Figure 2.3. Named Data Network Table Format [1]

2.1.3 Packet Naming

Interest packets have a selector space to provide the information about the content that needed, and a nonce space which is a random number produced by the consumer, as shown in Figure 2.2. A Data packet conveys the real content file, with an explanation of the Data and encrypted signature. The signature allows the consumer to confirm the trustworthiness of the Data packet. Subsequently, confidence in the Data packet is decoupled from how or where it was acquired. NDN built-in data security helps privacy from numerous viewpoints. Nevertheless, the signature on the routing protocol and control messages increases the security of routing conventions against assaults such as spoofing and tampering.

NDN names are new to the network and the router does not comprehend their significance. This empowers applications to characterize their own naming plans autonomously of the network. NDN names are structured hierarchically. For example, a document can be named in the form/uum/edu/cs/Alsamman/Thesis.doc in NDN where

“/" is the space mark between prefix name. This hierarchical arrangement has two benefits. Initially, applications can set the attachment and relationship among data components in the names; segment 1 of version 2 of this document can be named /uum/edu/cs/Alsamman/Thesis.doc/2/1. Secondly, it permits name aggregation, to improves the scaling directing framework. For instance, let the university be a self-ruling framework in NDN, it can then distribute the name prefixes as /uum/edu/ through directing conventions, like IP prefixes distribution in today’s Internet [1].

Like packets in IP packets, NDN networks provide the efficient way to retrieve Data. Data packet or an Interest packet can be lost, prompting the consumer to re-transmit the Interest if the Data has not been received after the anticipated RTT. Consequently, IP’s location structures for Data delivery in NDN packets convey content rather than locations. This essential distinction in designing packets has two significant effects on Data packet forwarding procedure. First, despite the name in an Interest packet, the directory forwarding is also included. Just as in today’s Internet, address location is utilized to lead the sending of the IP packet. An Interest packet may pass a replica of the demanded content at the halfway routers which results in getting the content faster if there is a drop in the way; while the IP packet only delivers the content to the requester (if not dropped along the way). Secondly, the Interest packets have not to carry address nor name to recognize the consumer to bring back the content. Rather, the NDN router follows different approaches with interfaces for each sent interest and utilizes this data by granting a matched Data packet back to the consumer. For this reason, NDN is built with an additional facility known as a forwarding plane.

2.1.4 Forwarding Plane

Every router in NDN is provided with a forwarding plane module, to establishe the connection from where and when to forward the Interest packets according to the details that is kepts on those data structures. In addition, a forwarding strategy is needed in every NDN router to maintain three significant data structures: Pending Interest Table (PIT), Content Store (CS) and Forwarding Information Base (FIB) (see Figure 2.3). The router's FIB in NDN is generally like the IP router FIB, with the exception in the architectural build that it contains name prefixes rather than IP address prefixes. This helps to forward the name prefix to different interfaces. The PIT section records the incoming interface(s) and name of each Interest packet(s), and also to which interface(s) the Interest packet(s) has been forwarded (like a history table). The CS gives short-term in-network storage (caching) for the coming Data packet.

Figure 2.4 shows the forwarding procedure in NDN. At the point when a router receives an Interest packet, it first looks in the CS for matching Data. The Data is instantly returned to the interface from which the Interest is coming. The router forwards it to look for a match with the name in the PIT entries. If the name prefix already recorded in the PIT, it may be a duplicate Interest that ought to be dropped, or the consumer retransmitted the Interest packet to utilize alternate outgoing interfaces (or an Interest from an alternate consumer requesting the same Data). This makes the PIT check the nonce of the Interest and adds the number of the incoming interface to the current PIT entry. This adequately builds a multicast tree for consumers asking for the same content at the same time.

However, if the prefix named of the Interest packet does not recorded in the PIT, it is recorded in PIT and moved to the FIB where it is handled by the forwarding strategy module. At this point, when a Data packet arrives, its name is used to check the PIT. If there is a match in the PIT entries, the router sends the Data packet to the interface(s) from which the Interest arrived and deletes the PIT entry. Subsequently, Data packets reliably take the reverse of Interest packets. In the event that no match is found, the

Data packet is cached in the CS or dropped. Every Interest packet also has a related lifetime placed by consumer, whereas entry of the PIT is evacuated if the Interest packet has not been fulfilled by its expiry time.

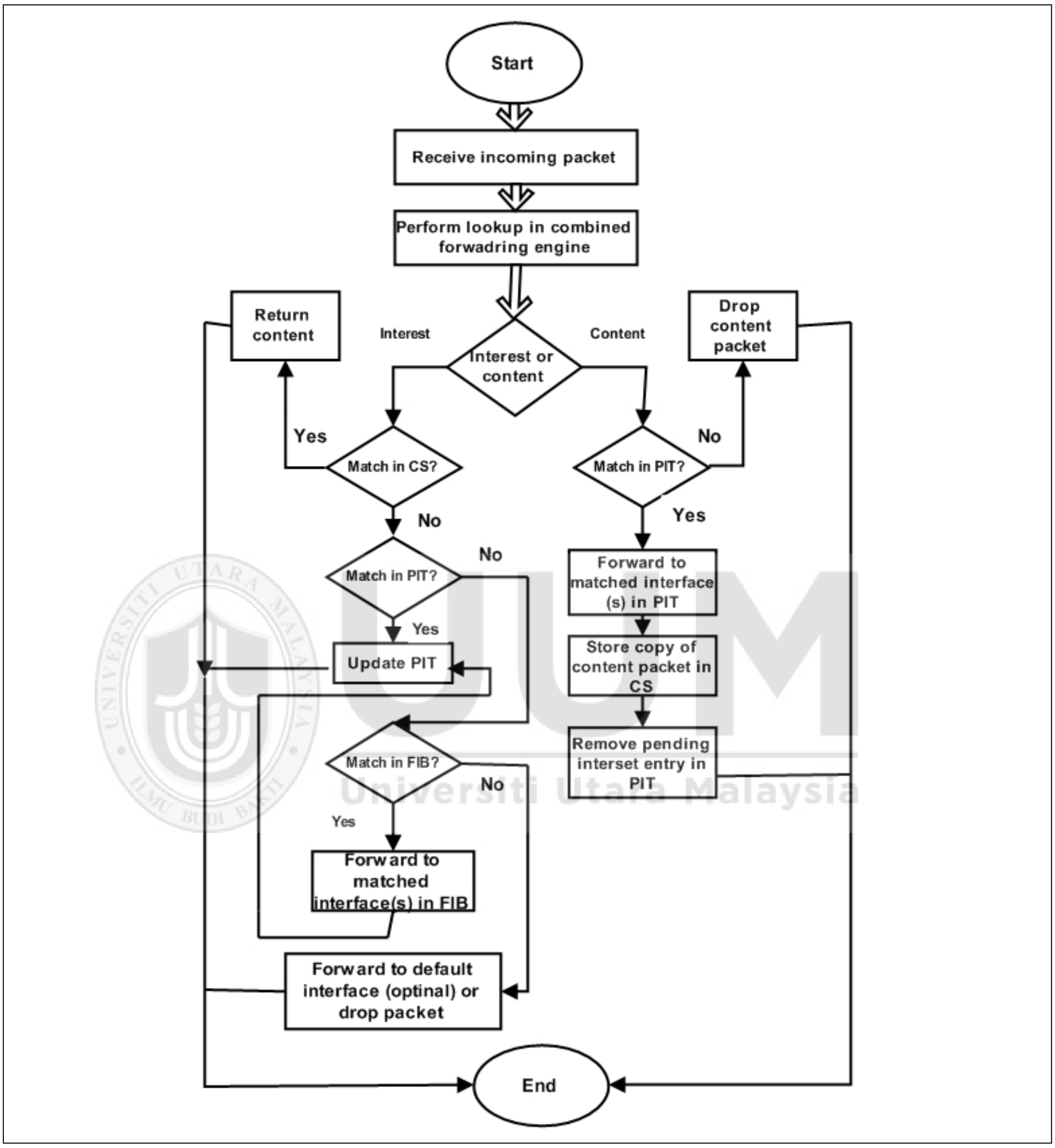


Figure 2.4. NDN Forwarding Plane [1]

Nonetheless, the NDN router may decide to store a Data packet in the CS because of its reliance on the caching strategy and the signature that makes it unique, so that the stored Data packet can be utilized to fulfil future Interests. In practice, the CS is equivalent to the IP router memory, but the Data packet is not reusable after it forwarded in IP router. The caching in NDN is usually helpful. For example, a Data packet may be cached by router A before it is dropped because of congestion or interrupts. The consumer will re-transmits the Interest packet and get the Data packet from router A without need to sent it to the real producer. This is seen as a promising advantage of the NDN.

Furthermore, NDN forwarding is loop-free because of the nonce field that each Interest packet has, jointly with the Interest prefix name to uniquely identify the Interest packet. The PIT in each router registers the nonce for each Interest packet entry. Consequently, if any arriving Interest holds the same name and nonce as a current pending interest which has already been forwarded, it is looped once again because that Interest packet cannot re-loop. Since Data packets take the reverse path of the Interests packets, because of that there is no looping. This empowers the router to retry a different path in Interest packet forwarding. For that, the retry has to be restricted in domain and time because, routers are not overall in charge of obtaining the content, and if routers on all the way perform retransmission, this might prompt the Interest's outstanding overhead.

Without Data packet loss, the Interest packet recovers precisely one Data packet on every interface, giving a balanced stream. When exchanging a Data packet in this stream equalization, the Data packet can give a feedback loop to each node on the forwarding path. This is basically like Acknowledgments (ACKs) in TCP.

2.1.5 Transport Control

While NDN solves Internet traffic problems by changing the principle of routing, this change also affects the network management from routing plane to forwarding plane. In addition, many issues have emerged in NDN architecture to make it applicable to the future Internet. These issues provide the motivation for researchers to address many shortcomings in order to realize its implementation. The research issues include cache management schemes at NDN routers, variable length names (prefix), routing and transport control, privacy, security and naming [26]. This research focuses on one of these issues, transport control, as it poses many challenges to make the NDN adapt to the future Internet.

The NDN architecture does not have a separate transport layer. It moves the functions of today's transport protocols up into applications, their supporting libraries, and the strategy component in the forwarding plane. The forwarding plane is built with the ability to detect failures (from links, nodes, and packets). For recovery actions, routing need not be performed continuously on FIB updates to improve the scalability and stability of the NDN routing plane [67]. For this research, transport control in NDN classified into forwarding plane and congestion control. Forwarding plane responsibility is to take the decisions needed for the Interest and Data packet forwarding inside the NDN router. Also uses as multiple forwarding option efficiently to chooses the best interface(s) to forward the Interest packet. On the other hand, congestion control, can control the congestion from consumer, router or both by hybrid consumer and router to control the congestion. Firstly, the consumer controls the flow by increasing the flow rate when receiving Data packets and reducing it when receiving NACK or when Interest times out. Secondly, the router side controls the flow by monitoring links and queues if there is congestion router send explicit notification or dropping the packets. Thirdly, a hybrid of consumer and router to control the congestion by letting the router to monitor and forward the incoming packet, when there is congestion router control it by sending packets to other path or by shaping and delaying them in the queue before sending to the consumer; see Figure 2.5.

Among other forwarding strategies deployed by the NDN router transport control are congestion control (as discussed in [13, 44, 68, 69]), adaptive forwarding (as discussed in [51, 70, 71, 72]), blind forwarding (as discussed in [73, 74, 75]), and aware forwarding (as discussed in [76, 77]). The transport control provides a resilient and reliable delivery at the application layer by monitoring and sustaining the status of the incoming Data packet (e.g., a given time threshold). If the consumer still wants the unsatisfied Interest packets, it re-transmits them after some threshold of time. In the router side, the transport control monitor controls the status of the transmission queue and forward adaptation [27, 12].

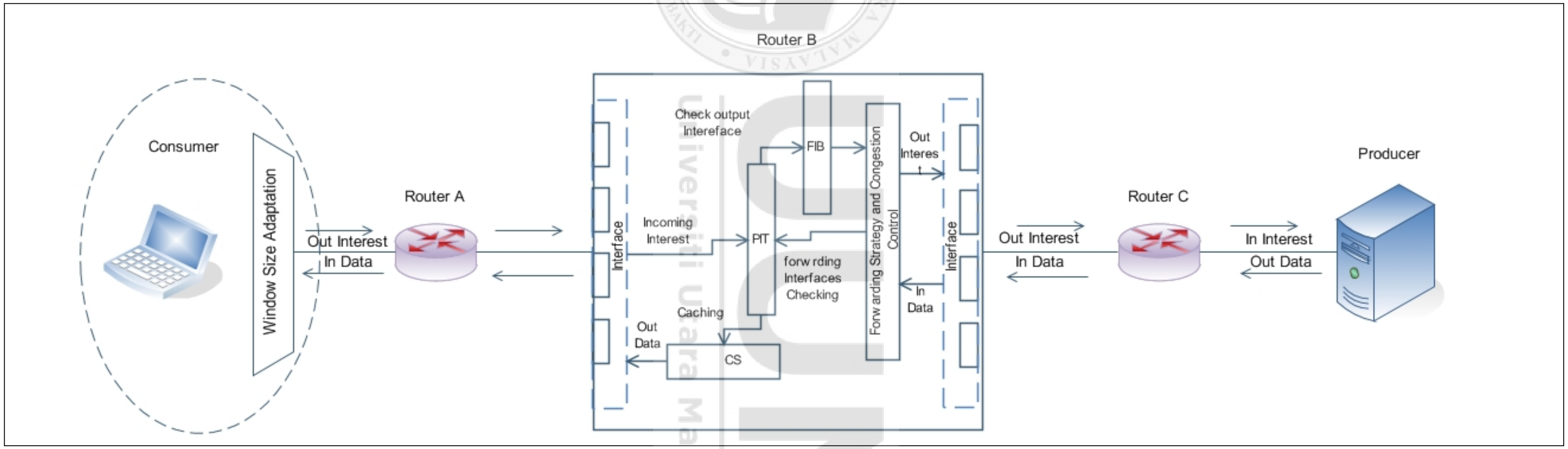


Figure 2.5. NDN Transport Control Structure

2.2 Transport Control Techniques

Earlier research on NDN generally concentrates on individual features, such as adaptive forwarding, in-network caching and naming, while avoiding the transport control that concerns overall network-wide resource utilization and receiver experience [12]. Due to different characteristics of the network architecture, one of the main challenges in NDN is to develop a global standard for transport control. Which nodes are not forced to choose a congestion control mechanism, as the receiver-driven transport control and built-in network caching of NDN architecture impact the overall system and receiver performance. NDN's architectural shift forwarding and congestion control mechanism to the router avoid buffer overflow, indicate early congestion detection and protect from misbehaving consumers.

In this section, transport control in NDN is reviewed based on techniques in proposed and published studies from 2012 to the present, the earlier date marking the beginnings of NDN (see Figure 2.6). The researcher focuses on developing a transport control in NDN because it is different from current Internet transport control, and each article proposes a different solution. Most of the proposed mechanisms concentrate on consumer-side congestion control and hop-by-hop Interest packets shaping. Some research concentrates on adaptive and intelligent forwarding strategy, and others investigate the whole NDN transport control as global optimization combining congestion control and dynamic forwarding.

Each technique proposed is discussed and analyzed in detail in the following subsections, from two perspectives. First, consumer-side control is either Timeout-based or Explicit Feedback control; see Tables 2.1 to 2.3. Secondly, router control is classified into hop-by-hop Interest shaping control and evolution algorithm forwarding adaptation see Tables 2.4 to 2.5. Each approach analyzed in several dimensions: title,

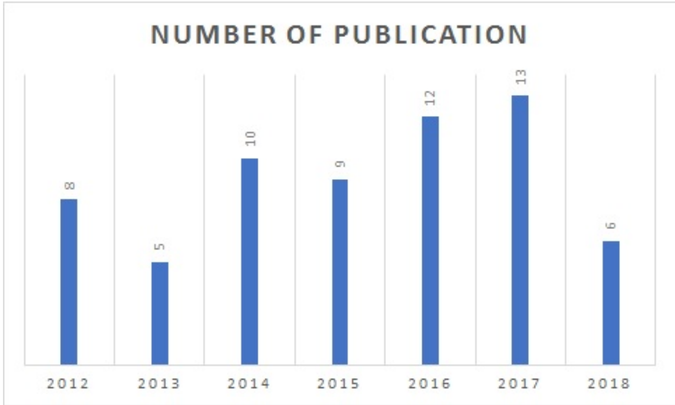


Figure 2.6. Related Work Statistic and Year of Publication

year of publication, congestion indicator, environment method, performance metric and contribution. The main characteristics and gaps of each mechanism in today's transport control are derived.

2.2.1 Consumer-side Control

Several strategies adapt TCP's IP control that operated using window-based or rate-based approaches. The proposed window-based mechanisms use timeout or delay to adapt the window size. Rate-based mechanisms use explicit notifications to adapt the sending rate user-side. These mechanisms and their functions are summarized below.

2.2.1.1 Timeout-based

In the timeout-based mechanism, the Interest packet rate is automatically regulated by a timeout update mechanism observing the RTT average delay. This timeout-based mechanism was among the first trials of NDN transport control research and, as shown in Table 2.1 is similar to TCP RTO mechanism. The challenge of this receiver-based method is how to adjust the Interest packet rate in the interface of the varying RTT.

There are different mechanisms to estimate the timeout. Jacobson's RTO estimation mechanism is the first type used to update the timeout mechanism and AIMD, except for adjusting some parameters. The authors in [31], adapt TCP congestion control to the consumer-based CCN setting, to provide fairness between competing flows. The authors in [78], correlate the performances of NDN using AIMD and TCP using AIMD and the analytical investigation, concluding that the throughput is better than NDN because of the in-network caching. The second type used the weighted average timeout estimator as in Interest Control Protocol (ICP) in [8] and [79]. The authors assuming that estimating the weighted average of the RTT measurements history would produce a reliable Interest packet control. Nevertheless, the latter type used the timeout as in CCTCP [33], predicting the place of Data packets by using an anticipated Interest packets mechanism and keeping individual timeouts for every expected source. As the predictive mechanism is more robust and performs better than the other two, the complexities of the design and implementation were significantly higher. The experiment in [80] on all three types of timeout-based was the definitive study.

Remote Adaptive Active Queue Management (RAAQM) [59] control of the Interest packet rate at the consumer side determines the minimum and maximum round-trip time delay, anticipating when the rate needs to be reduced. RAAQM proposes a single window and separate RTT calculations to control the multipath forwarding. It therefore assigns a unique congestion window of Interests with separate per-route RAAQM states to attach route labelling to each Data packet that carries a uniquely named route. This mechanism was improved in [63] by proposing a set of optimal active request forwarding strategies based on the number of pending requests; this is the basis of the plan for a fully distributed mechanism that runs at NDN routers. An On-demand Multi-Path Interest Forwarding (OMP-IF) [60] strategy uses multiple paths concurrently, based on RTT in each NDN router. Paths deployed with the OMP-IF strategy are based on node-disjoints as it recognizes several suitable interfaces from the available candidates provided by FIB. OMP-IF adds a table to the NDN router structure to keep and calculate the good interface and use it to forward any incoming Interest packets. Authors in [34] propose an analytical model to evaluate the impact of multipath forwarding strategies on the performance of ICN content transmission, whose congestion control follows a receiver driver. It pairs the loss base congestion control with AIMD consumer-side to control the Interest packet rate and take the pending interest in each path to employ the multipath forwarding strategy in each NDN router.

A multipath transport protocol at receiver in CCN [35] uses independent congestion windows management in consumer and calculates RTT for each path separately. An Entropy-based Probabilistic Forwarding (EPF) [81] strategy is proposed to achieve adaptive forwarding in NDN router and RTO in the consumer side. EPF uses a stochastic interface selection based on the combination of interfaces dynamic measurement like RTT, with bandwidth and static routing information by using Multiple Attribute Decision Making (MADM) mechanism. Authors in [36] aggregate congestion states of each sub-flow in consumer nodes by combining RTT of each sub-flow sent from a consumer to content sources over separate routes and control the forwarding based AIMD. Other mechanisms adapt the routing performance by proposing each path to assemble the content places and attach the content path by tagging the Interest packet headers, as in [82].

Table 2.1
Receiver-driven Perspectives

N	Title	Year	Congestion Indicator	Environment Method	Performance Metrics	contributions	QA Score
1	Transport-layer Issues in Information Centric Networks	2012	Time-out	An Implementation Based on CCNx	Good Put- Packet Loss	Proposed a transport protocol based on Timeout and AIMD Interest Window Control scheme [31].	3.5
2	AIMD and CCN: Past and Novel Acronyms Working Together in the Future Internet	2012	Time-out	Math Model	Throughput Gain- Caching Capacity	Proposed a congestion control mechanism based Time out and AIMD to avoid the starvation of less popular content downloads [78].	3.5
3	ICP: Design and Evaluation of an Interest Control Protocol for Content Centric Networking	2012	Weighting Average of Timeout	Math &CCNPL-Sim	Queue Delay- Window Size- Delivery Time	It is the first congestion control mechanism that take the weighting average of timeout to control the AIMD [8].	4.5
4	CCTCP: A Scalable Receiver-driven Congestion Control Protocol for Content Centric Networking	2012	Timeout	NS-3	Flow Completion Time	It anticipate the location of chunks to estimate the Timeout of each location and use AIMD Interest Window Control [33].	3.5
5	An Empirical Study of Receiver-based AIMD Flow-control Strategies for CCN	2013	Timeout	OMNet++	Transfer Time Ratio- Interest Retransmissions- Throughput	It used the Timeout and AIMD to control the interest rate in both consumer application and router forwarding strategy [80].	3
6	Multipath Congestion Control in Content Centric Networks	2013	RTT	CCNPL-Sim	Rate- Window Size- Estimated RTT	It proposed a Delay-based Probabilistic (RTT) Window Decrease mechanism that use AIMD to adapt interest rate in consumer side and to choose the available multipath in forwarding strategy in router [59].	4.5
7	An On-demand Multi-path Interest Forwarding Strategy for Content Retrievals in CCN	2014	RTT	OPNET	Average Content- Download Time, load Balancing Measure and Average Hop Count	Information table and face control is proposed in this mechanism as the table contain the different good interface and bad interface that identified by RTT of each interface and prefix. this mechanism use WRR to forward the interest for each path [60].	3
8	Multipath Forwarding Strategies in Information Centric Networks with AIMD Congestion Control	2014	Pending Interest-	PlanetLab test-bed	Receive rate and Congestion Window	It couple the loss base congestion control in consumer and forwarding strategy that take the pending interest in each path to utilize the multipath [34].	4

Table 2.1

Receiver-driven Perspectives (Continued)

N	Title	Year	Congestion Indicator	Environment Method	Performance Metrics	contributions	QA Score
9	Coupled Multipath Congestion Control at Receiver in Content Centric Networking	2015	Time-out.	CCNPL-Sim	Data rate, Fairness, Congestion Balance	It coupling consumer control and multi-path control using special parameter to differentiate between paths [35]	3.5
10	Efficient Multipath Forwarding and Congestion Control without Route-labeling in CCN	2015	RTT	Event-driven C++	Throughput- Average Content Delivery Time- Congestion Window	It aggregate congestion states of each sub-flow in consumer nodes by combining RTT of each sub-flows that sent from a consumer to content sources over separate routes and control the forwarding based AIMD [36].	3.5
11	An Entropy-based Probabilistic Forwarding Strategy in Named Data Networking	2015	Bandwidth and RTT	ndnSIM	Throughput, Load Balance, Drop Rate	It use a stochastic interface selection based on the combination of interfaces dynamic measurement like RTT and bandwidth and static routing information by using MADM mechanism [81].	3
12	Optimal Multipath Congestion Control and Request Forwarding in Information Centric Networks:	2016	Pending Interest.	CCNx prototype	Average Delivery time, Interest Rate and Split Ratio	It proposed an optimal interface selection per output and per prefix that take the pending interest in each face to calculate the forwarding rate [63].	4
13	PTP: Path-specified Transport Protocol for Concurrent Multipath Transmission in Named Data Networks	2018	Probing the network	ndnSIM	link utilization- Reaction Time- Fairness	It proposed two stage control first the probing stage to collect path information second stage consumer tag the interest packet with path information that router can use it for forwarding it. For control the congestion it use AIMD to increase that rate if received data packet and decrease if not [82].	4

2.2.1.2 Explicit Feedback Control

In [28, 30], a combination of an adaptive forwarding strategy is given an explicit feedback mechanism. The forwarding state is controlled in FIB and PIT in the first study to introduce employment of a NDN routers' datagram to create an exceptional and adaptive forwarding plane. These authors also proposed the first explicit feedback packets signalling called NACK. NACK is created by the congested router to notify the downlink router to decrease the Interest sending rate. They introduce a three-colour forwarding strategy, each colour describing the forwarding state of each interface. The Green interface has high performance; Yellow implies that the state is unknown; and the interface coloured Red is not working. Different metrics are used to assign the colour for each interface, such as routing preference, forwarding preference, RTT and number of incoming NACK packets.

AIM [83] extends the NACK method by forwarding it directly to the consumer in the reverse direction of the Interest packets. In Active Request Management (ARM) [84] NACK probability is produced based on the number of pending Interest packets, to prevent congestion. ARM is slow to adapt to the congestion as it uses AIAD consumer-side to control the Interest rate. Heterogeneous-Latency Adaptive Forwarding (HLAF) [48] monitors the routers' queue length and RTT concurrently to detect congestion and send the NACK to the consumer to adjust the forwarding rate. After the consumer in HLAF receives the NACK packet, it reduces the forwarding rate using the AIMD. QoS Forwarding Strategy (QoS-FS) [47] monitors, in real-time, ingoing and outgoing networks' links to estimate the QoS parameters cost, BW and RTT. It then combines them into the different decisions taken to determine when and which interface to use to forward an Interest. When QoS-FS fails to find a path to forward the Interest packet, it generates back a NACK signal to the downlink router or consumer using AIMD to reduce the Interest rate.

Stochastic Adaptive Forwarding (SAF) provides probability-based forwarding on a per-content/per-prefix basis. The extensive usage of multi-path transmission is the foundation of SAF's performance. SAF investigates the Interest satisfactory and unsatisfactory ratio on individual interfaces using a delay hop count threshold and updating the Forwarding Table (FWT) implemented by the mechanism to choose the best forwarding interfaces.

NDN Hop-by-Hop RCP (NHBH-RCP) [85] and Multipath-aware ICN Rate-based Congestion Control (MIRCC) [86] extend the use of NACK by enabling the middle nodes to update the available rate in each router, attach it to NACK and send it back to the consumer to regulate the rate. The Explicit Control Protocol (ECP) [46] extends the NACK packet by determining the Interest queue length weighted average; it describes them by sending three different NACK packets (Overload, Busy, Free) to detect the congestion. The consumer of ECP uses a MIAIMD (Multiplicative Increase Additive Increase Multiplicative Decrease) mechanism to regulate Interest packet rates according to the congestion notification sent by routers. The authors in [49] proposed a mechanism to solve the excessive rate reduction problem that happens with NACK packets by introducing three states for each interface and the content name. The states are Normal, Congestion and Check that each consumer does not reduce the rate as it receives NACK packets. Receiver Driven Performance-based Congestion Control (RDPCC) [79] considers real network performance like RTT and bandwidth. See Table 2.2.

Table 2.2
Receiver-driven Perspectives Using NACK

N	Title	Year	Congestion Indicator	Environment Method	Performance Metrics	contributions	QA Score
1	A Case for Stateful Forwarding Plane	2013	Interest Rate Limit	ndnSIM	Data Retrieval Time- Link Failure Probability- Link utilization	It design a spacial packet called NACK forward to consumer when congestion occur in NDN routers [30].	5
2	Active Interest Management for Improving Flow Completion Time in Named-Data Networking	2014	Average PIT Size	ndnSIM	Flow Completion Time	It monitors the average PIT size, and sends NACK packets based on statistical probabilities of PIT size. Consumers increases the interest sending window by one if no timeout or reduce by half if received NACK [83].	3.5
3	An Interest Control Protocol for Named Data Networking Based on Explicit Feedback	2015	Weighted Average, Interest Queue Length	ndnSIM	Interest Sending Rate-Throughput	It compute the weighted average length of Interest queue as congestion signal and feedback it to the receiver using NACK and use MIAIMD Interest control in receiver (three kinds of NACK congestion signals) [87].	3.5
4	A Rcp-based Congestion Control Protocol in Named Data Networking	2015	Interest Sending Rate Limit	ndnSIM	Average Flow-Completion Time- ate	Each router encapsulate the available rate to interest and data packets and consumer will adapt the rate base on the minimum rate available [85].	3
5	MIRCC: Multipath-aware ICN Rate-based Congestion Control	2016	Interest Sending Rate Limit	ns-3	Link Utilization- Content Download time - Interest Rate-	Attach initial rate value by the producer to data packet and each router update it until received by the receiver to set the rate [86].	4
6	An Explicit Congestion Control Algorithm for Named Data Networking	2016	Interest Queue Length	ndnSIM	Throughput - Delay	This mechanism is extended to [87] by adding an explicit feedback to the receiver using Data packet and control the rate using MIAIMD [46].	4
7	QoS-FS: A New Forwarding Strategy with QoS for Routing in Named Data Networking	2016	Cost, Bandwidth, RTT.	ndnSIM	Data Delivery Time, Hops Count	At each node consider three QoS parameters: cost, BW, RTT) as the mechanism choose the lowest RTT to forward the Interests with high BW [47]	3.5

Table 2.2

Receiver-driven Perspectives Using NACK (Continued)

N	Title	Year	Congestion Indicator	Environment Method	Performance Metrics	contributions	QA Score
8	AF: Stochastic Adaptive Forwarding in Named Data Networking	2017	RTT	ndnSIM	Average Interest Satisfaction Ratio, Average Cache Hit Ratio and Average Hop Count	It investigating the Interest satisfaction ratio on individual interfaces in certain delay threshold is used to update the FWT that used to choose the best forwarding interfaces [88].	4.5
9	HLAF: Heterogeneous-Latency Adaptive Forwarding Strategy for Peer-Assisted Video Streaming in NDN	2017	Monitoring Interest Queue	ndnSIM	Download Time-Forwarding Allocation-Playback Quality	It monitor the data queue length to detect the congestion and send NACK to downlink router for adjusts the forwarding rate. As each router control forwarding based on the explicit congestion signals (NACK) and RTT and consumer used AIMD to increased and decrease the forwarding [48].	4
10	Active Request Management in Stateful Forwarding Networks	2017	Average PIT Size	ndnSIM	Content Download Time-Throughput- Packet Drop-Queue Utilization	ARM prevents network congestion by drop interest packets and generate random early NACK packets corresponding to the interest packets based on the size of the PIT and use AIAD to increase and decrease the rate in router and consumer [84].	4
11	Congestion Control Avoiding Excessive Rate Reduction in Named Data Network	2017	Interest Rate Limit	ndnSIM	Interest Sending Rate-Link Utilization-Throughput	It use NACK to notified congestion and use AIMD to control the rate on receiver and router by proposed three states Normal, Congestion and Check to each router interface [49]	3.5
12	Performance-based Congestion Control in Information Centric Network	2017	RTT	ns-3	Fairness - Throughput	It propose a real performance algorithm by monitoring the RTT of received Data packets [79]	3.5

Using Data packets is the second explicit feedback control. CHoPCoP [13] was the first mechanism to propose an explicit notification using the Data packets to handle multiple-path and multiple-source situations. In CHoPCoP a router identifies congestion by observing the length of the outgoing Data queue. When the queue becomes congested or reaches a certain threshold, Random Early Marking (REM) (similar to RED and ECN mechanisms) is introduced to mark the Data packets. The consumer indicates congestion when on receiving marked packets and adapts its Interest rate equal to the number of marking Data packets received. The consumer's Interest rate control consists of two stages: the slow start stage (exponential growth) and then the congestion avoidance stage (AIMD-based). The congestion avoidance stage starts when the rate approaches the threshold, or when the network is congested. Therefore, more mechanisms monitor the queues in router interfaces to detect the congestion and mark the Data packets as in the Congestion Control Scheme (CCS) [89], Explicit congestion control based CUBIC (EC-CUBIC) [50], a Practical Congestion Control scheme (PCON) [51], and Window-based Congestion control (WinCM) [55]. EC-CUBIC detects congestion by monitoring the queue length and tagging the congestion in the Data packets to notify the consumer to use the CUBIC mechanism when the queue length is above a certain threshold. CCS uses the RED mechanism to monitor router queues to mark Data packets and send them downlink to adjust their sending rates using the AIAD mechanism. Similar to CCS, PCON and WinCM use the CoDel mechanism to detect and notify network congestion and the BIC mechanism to control the sending rate.

Authors in [90] take the PIT occupation as a congestion indicator, motivated by RED to measure the congestion and mark the Data packet to send it to the consumer. Flow-Aware Congestion Estimation (FACE) [91] predicts the number of incoming Data packets by using flow information to estimate congestion and report to the consumer to use the AIMD mechanism to slow down the Interest packet rate. In Novel Cooperative and Fully-Distributed Congestion Control (NCFCC) [52], the authors proposed a Co-operative and Memory-efficient Token Bucket (CMTB) mechanism, which monitors the buffer size of each intermediate node (hop-by-hop) to avoid congestion. NCFCC controls the Interest rate on the consumer side using Fully-Distributed Congestion Control (FDCC) to decrease the rate when the consumer receives a particular packet called Reduce Sending Rate (RSR) from routers. ECN-based Transmission Control [92] calculates the available bandwidth of specific flows and attaches it to the Data packet to control the Interest packet sending rates.

The ECN-based mechanism [92] adopted the idea from Software Defined Networks (SDN), where the controller collects network information in a domain. Thus, each router sends the information about the bandwidth of the attached link as well as the average Data size on each link to the controller in its domain. Routers also send the active forwarding entries to the controller, informing the controller about the network topology. See Table 2.3.

Table 2.3
Receiver-driven Perspectives Using Marking Data

N	Title	Year	Congestion Indicator	Environment Method	Performance Metrics	contributions	QA Score
1	An Effective Congestion Control Scheme in Content Centric Networking	2012	Average Data qQueue Size	OMNet++	Interest Sending Window- Queue Size- Throughput	Each router monitor the queue length status and informs the requester by setting Congestion Information Bits into the returned Data packet to adjust Sending rate in receiver using a proposed window adjustment algorithm [89].	4
2	A Transport Protocol for Content Centric Networking with Explicit Congestion Control	2014	Moving Average of Data Queue Occupancy	An Implementation on the ORBIT Testbed	Queue Size- of Timeout- Data Rate- Window Size- Throughput	The first transport protocol with explicit congestion control that use Random Early Marking (REM) to (mark data packets) and AIMD-based receiver Interest control and queue length control in router [13].	5
3	A Proactive Transport Mechanism with Explicit Congestion Notification for NDN	2015	Interest Rate Limit	ndnSIM	Link Utilization - Content Download Time -Bottleneck Queue -Interest Sending Rate	Receivers control the interest sending rates based on the explicit bottleneck bandwidth information returned back in the Data packet that router compute it base on the average RTT [92]	4
4	Inferring and Controlling Congestion in CCN via the Pending Interest Table Occupancy	2016	Moving Average of Data Queue and PIT Occupancy	ns-3	Content Download Time- Squared Coefficient of Variation- Number of Interest Re-transmission Data Chunk Packet Loss Rate- Queue Length	Uses the average occupancy of the PIT to estimate the anticipated data packet transmission queue length and sends explicit congestion notification signals to consumer that adapt AIMD to adjust the forwarding rate [90]	3.5
5	A CUBIC-Based Explicit Congestion Control Mechanism in Named Data Networking	2016	Queue Length	ndnSIM	Data Rate- Drop Packets	It proposed an explicit congestion control (mark data packets) to control the queue length in router and CUBIC-based receiver Interest control [50].	3.5
6	A Practical Congestion Control Scheme for Named Data Networking	2016	Queue Sojourn Time	ndnSIM	Rate- Forwarding Percentage- Congestion Window- Queue Utilization	In each routers interface mechanism compute the packet sojourn time in Interest queue to detect t congestion and change the forwarding to other interface or send explicit feedback (marking data) to the receiver to control the rate using Conservative Window Adaptation1 base on TCP-BIC [51].	5

Table 2.3

Receiver-driven Perspectives Using Marking Data (Continued)

N	Title	Year	Congestion Indicator	Environment Method	Performance Metrics	contributions	QA Score
7	Improving the Transmission Control Efficiency in Content Centric Networks	2017	Predicts the Future Queue Occupancy	Self build Simulation	Data Queue Size- CWND- Content Download Time- Throughput - Interest Unsatisfied Rate -Retransmission Rat	It propose Flow-Aware Congestion Estimation model (FACE), which estimates congestion based on flow information as well as queue utilization and notifies the congestion explicitly to consumer to adjust the rate using AIMD [91].	4
8	Novel Cooperative and Fully-Distributed Congestion Control Mechanism for Content Centric Networking	2017	Monitor Queue Occupancy	ndnSIM	Throughput- Packet Delay- Interest Retransmission	It proposed two algorithm one in router to monitor the queue size and send explicit congestion notification to consumer side that control the sending rate using AIMD concept [52].	4.5
9	A Rate-Based Multipath-Aware Congestion Control Mechanism in Named Data Networking	2017	Interest Rate Limit	ndnSIM	Link Utilization, Packet Drop, Forwarding Percentage and Forwarding Rate	Each router calculate the available Data sending rate for each flow and send it to consumer for adapt forwarding rate using Data packets or NACK [93].	4.5
10	WinCM: A Window-based Congestion Control Mechanism for NDN	2018	Queue Sojourn Time	ndnSIM	Delay - Throughput -Forwarding Percentage	In each routers interface mechanism compute the packet sojourn time in Interest queue to detect t congestion and change the forwarding to other interface or send explicit feedback (marking data) to the receiver to control the rate using Conservative Window Adaptation1 base on TCP-BIC [53].	4

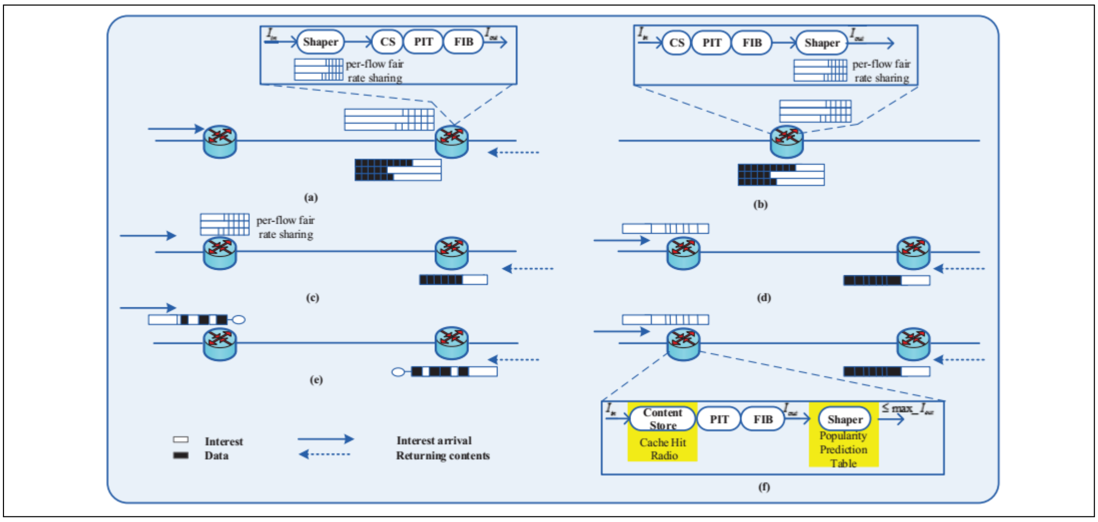


Figure 2.7. Illustration of simplified Interests shaping models [94]

2.2.2 Hop-by-hop Interests Shaping

NDN routers can proactively allocate the network capacity among different flows, anticipating the drop of Data packets due to queue overflow. Hop-by-hop shaping control is performed in NDN routers to avoid Data packet drop by regulating Interest forwarding. The shaping is done by monitoring the Data packet queue for congestion; the forwarding plan in the router will delay the Interest packet in router until the Data queue is no longer congested. Several hop-by-hop mechanisms are presented in Table 2.4, claiming that hop-by-hop Interest shaping is the best choice for NDN congestion control than consumer-based mechanisms. Figure 2.7 compares simplified Interest shaping models.

Hop By Hop Interest Shaping (HoBHIS) [38] was the first shaping mechanism in NDN to adopt the hop-by-hop mechanism by driving the queue length of Data packets to meet the a given objective value. This value is obtained by regulating the Interest packet rate. On the arrival of a Data packet in the output communication queue, the router calculates the Interest packet rate based on the occupancy of the Data queue, the available bandwidth and the RTT. The shaping time element is responsible for the calculation of the shaping delay that the Interest packets will have to satisfy. Meanwhile, when multiple conversations are received through the NDN router, the available queue size is distributed among all incoming flows.

Table 2.4
Hop-by-hop Perspectives

N	Title	Year	Congestion Indicator	Environment Method	Performance Metrics	contributions	QA Score
1	An Effective Hop-by-hop interest Shaping Mechanism for CCN Communications	2012	Queue Length	ndnSIM	Average Queue Size- Chunks Rate	It is the first shaping mechanism that use monitor the queue data length if it get above threshold it delay interest packet in the router and discard it if the data queue get congestion [38].	4
2	Joint Hop-by-hop and Receiver-driven Interest Control Protocol for Content Centric Networks	2012	Queue Length	CCNPL-Sim	Window Size	It maintain one virtual queue per flow, which refers to a single content retrieval and is identified by the name of the content. Each virtual queue is associated with a credit counter initialized to a maximum value, [40]	4
3	An Improved Hop-by-hop Interest Shaper for Congestion Control in Named Data Networking	2013	Interest Queue Length	ndnSIM	Queue Length- Window Size	Authors analyze the congestion issue mathematically by formulating it as an optimization problem to obtain the optimal shaping rate [41]	5
4	An Extended Hop-by-hop Interest Shaping Mechanism for Content-Centric Networking	2014	Queue Length	ndnSIM	Queue Length	This mechanisms is an enhancement of HoBHIS mechanism by adding a Tolerance mechanism to feedback the available rate to consumer that control the rate using AIMD [39].	4
5	Popularity-based Congestion Control in Named Data Networking	2014	Interest Rate-based	ndnSIM	Cache Hit Ratio- Interest Retransmit Rate- Flow Completion Time	It use shaping mechanism that delay the interest and forward it to up link based on pre-calculated content popularity as it use the caching mechanism to give the which content is popular [95].	3.5
6	A Traffic Aware Routing Protocol for Congestion Avoidance in Content-Centric Network	2014	Average Queue Length	OPNET	Throughput- Average Turnaround Time	It adopt the concept of AQM and RED mechanism to monitor the probability average queue length if get above threshold the router will delay the interest or drop it [96].	3.5

Table 2.4:

Hop-by-hop Perspectives (Continued)

N	Title	Year	Congestion Indicator	Environment Method	Performance Metrics	contributions	QA Score
7	Vip: A framework for joint dynamic forwarding and caching in named data networks	2014	flow rates and queue lengths	Simulation	Delay- Cache Hit- Arrival Rate	A systematic framework for jointing dynamic Interests forwarding by proposing Virtual Interest Plane to control the flow rate and queue length of the router and proposed a long side dynamic cache placement mechanism [97].	4
8	A novel multi-path traffic control mechanism in named data networking	2015	Interest rate-based	ndnSIM	Throughput, Waiting Interest, Fairness	It is combine the traffic control with multi-path forwarding strategy to control the rate between interfaces [65].	5
9	Hop-by-Hop Congestion Control for Named Data Networks	2017	queue Length	ndnSIM	Queue Occupancy- Received Data	A virtual queue for each prefix in each interface is created and monitor it's occupancy and if it exceed it fair share mechanism will send explicit packet to downlink to decrease it interest rate [42].	3
10	Parallel multi-path forwarding strategy for named data networking	2016	Pending Interest and RTT	C++	Execution times	It proactively splits traffic by us the weighted alpha fairness among different flows. It calculate the wight by taking the cache , pending interest, RTT and Data size of each flow [64].	4
11	A Congestion Control Method for Named Data Networking with Hop-by-hop Window-based Approach	2018	RTT	ndnSIM	Window Size- Throughput - Link Utilization- Queue Size	It use per-hop window size control in router by calculate the RTT with the neighbor to over come the variation of RTT that effect consumer control [43].	4

However, HoBHIS does not reflect the consumer side or shape the flow in the router, and if any congestion mechanism drops packets; the mechanism is therefore enhanced by adding a tolerance mechanism to feedback the available rate to the consumer [39]. The authors in [42] enhance the HoBHIS mechanism by proposing a virtual queue for each flow. A practical Hop by hop Interest Shaping mechanism (HIS) proposed in [41] takes into account the unique relationship between Interests and returning Data in NDN, and is shown to perform proportional fairness between two-way traffic. The authors investigate the congestion issue mathematically by formulating it as an optimization problem to reach the optimal shaping rate. They present a practical shaping algorithm based on these solutions and independent of traffic flows, to be executed at the output interface of NDN routers. This also combines hop-by-hop interest shaping with an AQM backpressure mechanism as a more viable option for NDN congestion control. Hence, this mechanism combines consumer control to adjust the Interest rate if the shaping in the router cannot adjust it. It adapts the multipath forwarding mechanism to improve the throughput. A Parallel Multi-Path Forwarding Strategy (PMP-FS) [64], the first parallel multipath mechanism in NDN, adopts HIS [41] to avoid congestion and shape the traffic. PMP-FS proactively divides traffic by determining how the multiple routes will be used to consider in-network caching and Interest aggregation characteristics to achieve weighted alpha fairness among different flows. For this, it employs a mechanism used in Software Defined Networks (SDN) [98] to calculate the weight of each interface by using the cache, pending interest, RTT and Data size of each flow.

The authors in [40] enhanced the ICP mechanism by joining hop-by-hop Interest shaping control in intermediate routers, calling it the Hop-by-hop and Receiver-driven Interest Control Protocol (HR-ICP). HR-ICP starts by checking the link bandwidths and building a virtual Data queue and credit counter for each incoming flow at each output interface in intermediate routers. The counter of each flow is first initialized by the maximum value allowed to transfer without any delay. HR-ICP then calculates the max-min fair rate for each flow by dividing the link bandwidth into two categories: bottleneck and non-bottleneck. The non-bottleneck flows take their bandwidth requirement and the rest is divided equally between the bottleneck flows. The counter of the shaping rate is increased when the assigned bandwidth in the downlink increase and is reduced with the router forward Interest packets to uplinks. In Chunk-switched Hop Pull Control Protocol (CHoPCoP) [13], as mentioned previously, when the router indicates congestion it sends the consumer an explicit notification by marking the Data packets. If the consumer does not decrease its Interest rate when it receives the notification, CHoPCoP proposes a shaping mechanism in the router called the Fair share Interest Shaping (FISP) algorithm. FISP delays Interest packets on the up-link Interest queue, similar to HoBHIS [39], but periodically calculates the total occupancy needed by each flow by taking the number of Data packets present in the Data queue and the number of Data packets that will come into play for the Interest packets sent by the router. When the number of packets in the Data queue exceeds a certain threshold, FISP delays the Interest packets whose Data queue exceeds its fair share.

Popularity-based congestion control is proposed in [95], using pre-calculated content popularity stored in the NDN router cache to shape the Interest packets. The authors assume that the more popular the content, the greater the use by the downlink router. The mechanism calculates the available bandwidth of up-link reverse direction to shape the Interest packet rate. When a router's Interest packet sending rate is higher than the permitted maximum, it will forward the next Interest packets to the next higher rank listed in the Popularity Prediction Table (PPT). The Virtual Interest Packets (VIPs) [97] framework employs a virtual control plane operating on VIPs at the data object level, and an actual plane handling Interest Packets and Data Packets at the data chunk level. The virtual plane facilitates the design of distributed control algorithms operating on VIP content popularity. VIP aims at yielding desirable performance in terms of particular network metrics, by taking advantage of local information on network demand. The flow rates and queue lengths of the VIPs result from the control algorithm in the virtual plane.

In addition, a Traffic Aware Routing Protocol (TARP) [96] for congestion avoidance in CCN uses the concept of AQM and RED. TARP employs two methods. The first processes the Interest packet based on average queue length probability, when the average queue length is greater than the minimum threshold and less than the maximum. The second method discards the Interest packet when the average queue length is greater than the maximum threshold. The Multi-path Flow Control (MFC) mechanism [65] calculates the fair rate for each flow in the NDN router and attaches it to the Data packet to send it downlink. When the router receives the Data packet from the uplink it decreases or increases the forwarding rate accordingly. MFC distributes the incoming flow to different paths based on each path's available rate as received from its incoming Data packets. Authors in [43] proposed rate-based mechanism called the Hop-by-hop Widow-based Congestion Control (HWCC) method. In HWCC, the Interest window size is determined for an individual hop in each flow, so that Data packets can effectively use the network bandwidth. HWCC introduces a Hop-by hop Acknowledgment (H-ACK) packet and a queue (Interest queue) storing Interest packets while the window is closed. An H-ACK packet reports the reception of continuously transferred Interest packets. After a router (or a consumer) has sent Interest packets up to the window size, it waits for an H-ACK packet. Upon receiving this, the router determines the next window size and sends Interest packets within the new window size.

Table 2.5
Evolution Algorithm Perspectives

N	Ref	Year	Congestion Indicator	Environment Method	Performance Metrics	contributions	QA Score
1	Probability-Based Adaptive Forwarding Strategy in Named Data Networking	2013	RTT	Ns3	Average Delay and Effective Transmission Ratio	It adding probing packet to measure the link delay and using Ant Colony Optimization to selects face which probabilistic has the least delay [99].	4
2	A Neural Network Based Congestion Control Algorithm for Content-Centric Networks	2014	Packet Drop	ndnSIM	Queue Size- Packet Drop	It is use the neural network to predict the congestion by detect the packet drop or if link capacity is low and will change forwarding to other link or it will inform the consumer by delay the interest packets [100].	3.5
3	ACCPndn: Adaptive Congestion Control Protocol in Named Data Networking by Learning Capacities Using Optimized Time-Lagged Feedforward Neural Network	2015	Rate-base	ndnSIM	In Data- Drop rate- Interface Load- Cache Hits	It use two phase framework to control the congestion. The first phase being adaptive training using Particle swarm optimization, Genetic algorithm and Time Series Forecasting and the second phase using fuzzy avoidance in each interface [101].	4
4	Intelligent Forwarding Strategy Based on Online Machine Learning in Named Data Networking	2016	RTT.	ndnSIM	Throughput, Load Blance, Finish Time and Drop rate	It integrates online machine learning method into the optimization of interface probabilities that rank interface based on RTT [102]	3
5	Ant-Colony Optimization Based QoS Routing in Named Data Networking	2016	Pheromone Information	Matlab	Data Packet Delivery Rate, Delay Change and Cost Change	An ant-colony optimization method is adopted to enable the determinate and stochastic searching method to make selection in Pheromone Table that combined the local and global pheromone information [103].	3.5
6	AC-QoS-FS: Ant Colony Based QoS-aware Forwarding Strategy for Routing in Named Data Networking	2017	Bandwidth and RTT	ndnSIM	Data Delivery Time, Hop Count, Dropped Packets, Hit Ratio	It use the ant clone concepts of both forward and backward delay and bandwidth to rank interfaces [104].	4

Table 2.5

Evolution Algorithm Perspectives (Continued)

N	Ref	Year	Congestion Indicator	Environment Method	Performance Metrics	contributions	QA Score
7	A Quantified Forwarding Strategy in NDN by Integrating Ant Colony Optimization Into MADM	2017	Pending Interest and Pheromone Information	ndnSIM	Throughput, Packet Loss and Load Balance	It use the Ant clone optimization (ACO0 and multi-attribute decision making (MADM) to rank interfaces base on pending interest and pheromone information [105].	4
8	A Game Theoretic Framework for Congestion Control in Named Data Networking	2017	Packet Drop	ndnSIM	Interest Shaping Rate- Link's Utilization- Content Download Time- Data / Interest Discard Rate	It use the game theory to build the congestion control mechanism with objective to allocate a proportional fair resources among flows by taking flow characteristics [106].	4
9	Interest Forwarding in Named Data Networking Using Reinforcement Learning	2018	RTT and Caching Popularity	ndnSIM	Network Overhead, Cache hit	It employs an online learning algorithm, reinforcement learning using the random neural network, to rank the forwarding interface based on RTT and caching popularity [107]	3
10	A Novel Forwarding and Routing Mechanism Design in SDN-based NDN Architecture	2018	Cost, Bandwidth, RTT	Matlab-simulator	Convergence Behavior, Fitness of Path, Bandwidth Utilization, Number of Hops	It apply the SDN architecture prospective as Interest packets can be forwarded through an effective centralized routing strategy supported with genetic algorithm that take the resource consumption and network load to make routing decisions during packet forwarding [108]	4
11	ACCP: Adaptive Congestion Control Protocol in Named Data Networking Based on Deep Learning	2018	Weighted Average Length of Interest Queue	ndnSIM	Queue size- Data Rate- Window Size	First phase it proposed a deep learning prediction model used to predict the average queue length of each node. secondly indicate network congestion by outcomes of first phase and explicitly return Modified NACK to consumer to adjusts sending rate of Interest packets [109].	4

The authors in [99] proposed a forwarding strategy based on Ant Colony Optimization (ACO) called Probability-based Adaptive Forwarding (PAF). PAF take the Interest time-out to rank each interface using a statistical model. It adapts ACO to select the least delay interface to forward the Interest packets. It proposes a probed packet to calculate each interface RTT by sending prob-Interest and receiving prob-Data. It then reconstructs the router FIB to add the probing probability calculated for each prefix in the different router interfaces. A Probabilistic Binary Tree-based Forwarding strategy (PBTF) [102] abstracts the forwarding process as a path selection process traversing from the root node to the leaf node, providing theoretical support for machine learning and reducing the complexity of the forwarding process. It also prevents the convergence into limited local optimal solution by adopting the idea of simulated annealing. When PBTF receives an Interest that needs to be forwarded, the corresponding alternative interfaces are first extracted according to FIB. A binary probabilistic tree is then built with all the interfaces and calculates the RTT of every interface to choose for next forwarding. The interface with the smallest RTT PBTF will increase the value of the non-leaf node of this interface. The selection process is conducted from the root node to the leaf node according to the newly generated probability tree.

An ant-colony optimization-based QoS forwarding strategy [103] is proposed in combination with the node design method and NCE strategy of the SoCCeR algorithm [110]. The algorithm adopts the Neighbour Cache Table (NCT) to store the cache information of the neighbours' routing nodes and creates a Pheromone Table (PT) for each content producer. By using the NCT and PT the mechanism calculates the QoS performance (RTT) of each interface. Another ant colony-based QoS-aware forwarding strategy (AC-QoS-FS) [104] proposes making full use of both forward and backward ants to rank interfaces, taking Interest and Data packets as forward and backward ants' probes. AC-QoS-FS relies on ants measuring the real time network QoS parameters of the path they traversed from the data-source (producer) to the data requester (consumer), using these measurements to compute the amount of pheromone to be deposited in order to select the best interface to forward the incoming Interests. When arriving at each node of the path, an ant deposits a certain amount of pheromone for the corresponding incoming data interface, according to the measured QoS parameters of the path. The amounts of pheromone are used for ranking interfaces in order to select the best one for forwarding the incoming Interests. Ant Colony Based Extensible Forwarding (ACEF) strategy [105] formulates and approaches Interest forwarding as a multi-attribute decision making (MADM) problem. ACEF integrates an ACO algorithm and the state information of pending Interests, and employs a method named maximizing deviation to automatically allocate each attribute's weight during the Interest forwarding process. ACEF adopts pheromones and the number of Pending Interests (PI), which is associated with a given Interest and particular interface. It abstracts an Extended Information Table based on pheromones and the number of Pending Interests. It introduces the MD method to calculate the weight of each metric, obtains the real-time availability of the interfaces, and then decides which interface is the best candidate for a certain Interest packet.

Adaptive Forwarding Strategy using Reinforcement Learning with the Random Neural Network (NDNFS-RLRNN) [107] employs an online learning algorithm for reinforcement learning using a Random Neural Network (RNN) to forward Interest packets. In NDNFS-RLRNN, the RNN is created for a prefix in the FIB and its creation is triggered by a new Interest for the corresponding prefix. In its initial state, the RNN only knows of the routing preferences for its prefix and is yet to be updated by packet delivery measurements. As the goal of NDNFS-RLRNN is to minimize the delay for retrieving contents, so the RNN estimates the reward using the RTT values. NDNFS-RLRNN also adopts Interest NACKS when a router can neither satisfy nor forward an

Interest. An extension to the general SDN-based NDN architecture [108] designs new data structures in the controller and router, so that the controller can maintain global information of the local and adjacent network. In addition, the router can record the configuration information from the controller. The Interest packets can be forwarded through an effective centralized routing strategy rather than being flooded and can all be matched, to avoid needless abandonment and thus guarantee a higher response rate. A QoS multi-constrained routing algorithm is applied in the proposed forwarding mechanism with consideration of both resource consumption and network load balancing based on an improved GA algorithm, which is carried out in the controllers to make routing decisions during packet forwarding.

The authors in [100] use the drop occurrence prediction in a link as a signal to detect and avoid congestion. The Multilayer Perceptron (MLP) neural network is used to dynamically predict the occurrence of drop on each link in the routers. For this, each router in the network collects statistical information from the available traffic on the amount of each link connected to it. If the neural network detects that the likelihood of dropping packets on the link is low, then the Interest is transmitted in accordance with previous strategy. If the router predicts a drop on the link then it uses another link to forward the Interest. If the congestion is predicted for all the paths, the next Interest packet will be delayed and consumers alerted to reduce the Interest rate. Adaptive Congestion Control Protocol in Named Data Networking (ACCPndn) [101] has two phase frameworks for congestion control: adaptive training and fuzzy avoidance. Adaptive training tries to forecast the rate at which entries are added to the PIT table to take as a congestion indicator and predict the coming Data packets. This adaptation is done by three algorithms: Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and Time-lagged Feedforward Neural network (TLFN). In the fuzzy control system each contributing router applies three types of control per interface:

readjust Interest packets rate, affect forwarding strategy in the downstream, and set default configuration. Authors in [106] present a game theoretic framework for flow rate control in NDN based on the concept of Nash bargaining solution from cooperative game theory. They concentrate on shaping the flow in the router only by proposing a distributed flow-aware hop-by-hop congestion control mechanism on an analytical basis. This hop-by-hop congestion control mechanism uses proportional fair resource allocation among concurrent flows by taking into account the flow average response time, output link capacity, and the number of active flows. The Adaptive Congestion Control Protocol (ACCP) [109] is an explicit congestion control protocol based on two phases of deep learning. The first phase is adaptive training, employing the idea of [101] which predicts the increase in the PIT table of the routing node. Secondly, a Time Series Prediction model based on Deep Learning (TSPDL) predicts the time series data, trained by a Deep Belief Network (DBN) to study low-dimensional features. The low dimensional feature of time series data is used to train the conditional restricted Boltzmann machine time series model based on the Gaussian process (GCRBM) which predicts the later time series data. According to the prediction result of the first phase, ACCP estimates the level of network congestion through the average queue length in each router, and then the congestion level is explicitly feedback to the consumer. The consumer adjusts the sending rate of Interest packets to control the sending rate of Data packets using an Exponential Increase Addition Increase Multiplication Decline (EIAIMD) algorithm.

2.3 Critical review

The delay-based mechanisms listed in Table 2.1 has difficulty in detecting congestion as they use a single RTT. The deterministic window decreases are sensitive to RTT estimation errors, which frequently occurs in NDN due to inaccurate RTT measurements. This problem is worsened by the absence of delay samples during congestion

[51]. Additionally, an Interest packet has a specific lifetime. As the mechanisms in Table 2.1 uses the RTO to control the rate. The Interest lifetime may expire due to congestion, or its duration may be shorter than the network delay, resulting in packet loss and re-transmission of the Interest packet. The mechanisms that rely on it will consequently be affected. The late loss recovery problem is inherent in delay-based congestion control mechanisms because there is no way to discard PIT entries in NDN routers when a Data packet is lost. However, waiting for the expiry of the Interest lifetime in order to resend the Interest is not appropriate for delay-sensitive applications that require high throughput and minimum delay [30]. Furthermore, consumer control is not sufficient to detect available bandwidth in the network and ensure fairness between flows [10].

Receiver control expands the mechanisms shown in Tables 2.2 and 2.3 by proposing an explicit feedback mechanism in each hop-by-hop router. In Table 2.2 mechanisms use NACK packets as proposed by [30] to notify the receiver of congestion. Otherwise, as the NDN forwarding plane sends all received Data packets to all interfaces requested by the Data packet, the mechanisms in Table 2.3 take advantage of this by marking the Data packet when detecting congestion. It is better to mark the Data packet than send a NACK packet as by marking Data packet all downlink routers and the consumer will be notified at once, but by sending a NACK packet the router needs to send NACK to each downlink separately. On the other hand, for congestion detection on the router side, some mechanisms still use RTT to detect congestion [47, 88, 79] and they still face the RTT variation because of the multi-homing content.

Other mechanisms use queue length in detecting congestion [87, 46, 48, 89, 13, 50] to overcome the RTT variation, although they do not consider queue size variation in the network that affects delay-sensitive applications. Different mechanisms [83, 84, 90] take PIT size as a congestion indication, but this does not consider fairness between flows, as PIT acts as a table to all connected links whether with high or low bandwidths, and also Interest aggregation. In addition, the mechanisms in [85, 49, 92, 93] do not mark the Data packet to which they add the rate limit; as the receiver adapts the sending rate based on this, the sending rate to consumers in each Data packet takes more time, power and router CPU overhead, as the routers calculate the available rate every time they receive a Data packet.

Some mechanisms [51, 53] use Controlling Queue Delay (CoDel) to overcome the queue size variation and buffer bloat, but in terms of fairness this needs to be enhanced by combining the CoDel algorithm with a scheduling algorithm such as Fair Queue [56, 57]. Nevertheless, the mechanism that depends on received rate from uplink to adapt the forwarding has to wait for the notification, and if there is any loss of notification packets, the mechanism will misbehave. This delays the forwarding reaction and affects link utilization.

All these mechanisms rely on the consumer side to take action and only monitor the congestion in routers without taking any action to delay the Interest packets in routers. Thus, if there is any congestion in the router, these mechanisms send directly to the consumer or downlink to decrease the Interest packet rate. The feedback generated hop-by-hop will therefore be excessive. Some of these mechanisms attempt to solve the problem of uplink congestion by forwarding interest to another best interface [30]; others [86, 51, 52, 93, 53] use multipath forwarding to distribute rate pressure to different interfaces, but do not process the congestion inside the router. The consumer therefore receives numerous congestion notifications that affect the stability and efficiency of the network.

Furthermore, as NACK uses feedback congestion and consumes the Interest packet in the downlink, the mechanisms marking Data packets do not consider the Interest aggregations. When the Data packet is received by the NDN router, it is forwarded to all downlink routers which request the Data; when the router receives the marked Data, it decreases its rate even if some routers are not congested, affecting link utilization. Most of these mechanisms adopt AIMD, AIAD and MIAIMD congestion avoidance mechanisms that seriously reduce the ratio of stability and link utilization, affecting the performance of the network. The authors of [51, 55, 50] adapt BIC and CUBIC congestion avoidance mechanisms to overcome severe fluctuations. Nevertheless, these mechanisms still suffer from under-utilization of high-speed bandwidth and short distance, specifically when the buffer size is small [111].

The hop-by-hop rate shaping mechanisms proposed in Table 2.4 to control the forwarding rate in routers divide the queue in the router into Interest queue and Data queue monitoring. The mechanism in [38, 42, 96] monitors the Data queue; if its length reaches a certain threshold the mechanism delays the Interest packet in its queue until the Data queue falls below its threshold, or it will be dropped. However, the Interest drop in these mechanisms will increase the Interest re-transmission in the network and affect the scalability of the network. Receiver collaboration is therefore needed to overcome the dropping of Interest packets as these mechanisms [39, 41, 40, 13] start by delaying the Interest packets if there is congestion in the Data queue; if the router cannot handle the congestion it sends feedback to the consumer to reduce its Interest rate.

The mechanism in [95] calculates the available rate for each flow based on content popularity, other uplink notification shapes the forwarding rate or sends it in multipath [65]. Even though [64] adopt the mechanism in [41] to shape the congestion and adapt the SDN mechanism to forward Interest in parallel paths, it still depends on RTT and queue length. None of these mechanisms is independent as they need to know the bandwidth, queue size and the delay of the links in advance to control the forwarding rate; these parameters are not applicable in real networks [51]. Furthermore, none of the shaping mechanisms that uses queue length to indicate congestion considers the variation of queue size in routers, affecting delay-sensitive applications.

Table 2.5 discussed several evolution algorithms used to indicate congestion by calculating the router and network characteristics every time they need to forward. Several mechanisms [99, 102, 104, 107] suffer from multi-home content as they depend on RTT to indicate congestion. The mechanisms in [100, 106] indicate congestion by dropped packets, increasing re-transmission in the network that affects link utilization. New packets added to NDN architecture prompt the network to learn the available bandwidth and congestion [103, 93]. However, these calculations consumer router CPU space and power even in making the forwarding decision. The prompting packet used in evolution algorithms check network characteristics add more overheads to the network.

In summary, the deterministic window decreases are sensitive to RTT estimation errors, which frequently occurs in NDN due to inaccurate RTT measurements. Therefore, RTT measurements or any delay base calculation are not suitable to control the rate and congestion in the NDN [41, 66, 51, 55]. That guides the researcher to depend on the rate base control, by detect congestion in the router side and send explicit feedback to the consumer side. However, these changes move the challenges to the router side. In the router side, some mechanisms still use RTT to detect congestion while other mechanisms adopt several AQM schemes. The mechanisms that adopt RTT measurement in router increase the complicity not solve the problems. While the mechanisms that adopted AQM schemes overcome in-path issues, but other challenges have adopted with them like bufferbloat different size of queues that consider some of the issues faced by the adopted schemes in current networks [56, 57, 58].

Nevertheless, those mechanisms adopt the queue length indicator that lack of controlling the queue size variation that affects delay-sensitive applications. However, with indicate the congestion in the router side and sends feedback packet to consumers that still managed the forwarding rate and it takes time to respond to the congestion [41, 26, 52]. On the other hand, because the Interest packet aggregation in the NDN router is only adapting one explicit feedback increase the number of feedback to consumers and affect the rate stability. Therefore, hop-by-hop controlling mechanisms have proposed to overcome the delay in responding and even to control the forwarding inside the router by sending the Interest packet to other paths when there is congestion. Even more, these mechanisms shaping the forwarding rate by delaying the Interest packets in the router if there is any congestion in the uplink paths [41, 39, 26]. However, these mechanisms adopted the queue lengths measurement that lacks on controlling the queue size variation that affects delay-sensitive applications [51, 55]. Nevertheless, most of the mechanisms mentioned above not use a first in first out scheduling scheme to send to packets over interfaces that not appropriate to ensure the fairness between flows. Also, their forwarding control depends on configuring values or came from other uplinks that delay any action to prevent packets lost or congestion [66, 56, 57, 10].

2.4 Theories Pertinent in This Study

Two major theories are adopted in this research: Queuing Theory and Scheduling Theory. The following sub-sections introduce them in the context of this research.

2.4.1 Queuing Theory

Queuing theory is the key analytical modelling technique used for computer systems performance analysis [112]. It deals with stochastic models that depict the transformation of the subscribers' random flows during servicing by servers. In this way, Queuing theory is significant whatever the concept of the queue: breakdown, waiting or loss [113]. At this point, a Queue Management algorithm [58] is the process by which a router chooses when to drop a packet and which packet should be selected for dropping at its output port when it becomes congested. Queue management algorithms attempt to approximate fairness by appropriately dropping packets in order to minimize network congestion and keep up reasonable queue lengths. One case of a queue management algorithm is the Active Queue Management (AQM) (as discussed by Thiruchelvi and Raja in [114]; also discussed in [58, 115, 116]), it tries to balance congestion control at the endpoints to avoid packet dropping.

Characteristically, AQM is a proactive congestion control mechanism, where the data are sent by a network node to the sources if early congestion is detected. The data can be sent expressly as Explicit Congestion Notification (ECN) (as discussed in [117]) marks, or verified by packet drops [118]. When congestion increases, the AQM scheme intensifies its feedback to the Transport Control Protocol (TCP) endpoints, i.e., by dropping or marking more packets. The sources, in response to these congestion notifications, decrease their data transmission rates in order to avoid queuing overflows and reduce the losses that can enhance the flow control. Subsequently, the AQM must readily detect congestion and give quick and compelling feedback to the sources. Generally, the queue management mechanisms consist of three parts, namely:

- The congestion indicator.
- The congestion control function

• The feedback method.

The congestion indicator is utilized by the queue management mechanism to decide if there is congestion, whereas the congestion control chooses what needs to be done if congestion is identified. The feedback method is the congestion flag which is used for alerting the source to adjust its transmission rates. According to Thiruchelvi and Raja in [114], AQM can be classified into three families, namely queue length-based (e.g., Random Early Detection (RED) [119], Stabilized Random Early Drop (SRED) [120]), rate based (e.g., Adaptive Virtual Queue (AVQ) [121], Stabilized Adaptive Virtual Queue (SAVQ) [118]) and delay based (e.g., Controlled Delay(CoDel) [122].

CoDel is the latest AQM, proposed by Nichols and Jacobson in 2012 [122]. It was designed to solve the full buffer problem, "bufferbloat", in networks by limiting the packets delay in the routers queue. CoDel aims to increase the network overall performance by decrease the packet loss and delay of the flow while an increase in the throughput and the link utilization. According to [123], it has significant characteristics that make it better than other AQMs, such as:

- Parameterless: no pre-configured parameters required.

- Treating differently good queues (the queues that drains as fast as possible) and bad queues (the queues that get fills up at the transmission rate).

- Queue delay is controlled regardless of RTT delay and traffic load.

- Maintaining dynamically changing send rate without any effect on link utilization.

- Simple to implement in real router.

CoDel can be counted as one of AQM delay-based mechanisms because it employs packets sojourn time rather than packets arrival rate or queues length as a congestion indicator. Packets sojourn time is the time that packets spend inside the queue. Sojourn time can be calculated by stamp the enqueue time to each incoming packets and subtracting the enqueue time with dequeuing time to calculate the sojourn time for each packet independently. If the sojourn time is higher than a pre-defined target, CoDel initiates a timer for dropping the packets. This drop will happen only if the sojourn time is higher than the target and the packets in the queue have not exceeded the queue length. CoDel has two important parameters, target and interval, which must be configured carefully for better performance. However, these parameters are given fixed values, chosen based on many simulations and experimental results, [124] as follows:

- Target: constant 5ms (acceptable queue latency)

- Interval: constant 100ms (in worst case of RTTs)

CoDel has shown better results in many comparisons with previous AQM schemes. [125] compared CoDel with RED and Adaptive RED (ARED), and concluded that CoDel is independent of queue size, rate measurements, drop rate and RTT delays; they showed that CoDel has better performance in link utilization, queue length, and drop rate. According to [126], CoDel performs better than drop-tail and RED algorithms in terms of queue delay, link utilization and packet drop. In terms of fairness, CoDel is considered better than some of the RED variants, but it needs to be enhanced by combining the CoDel algorithm with a scheduling algorithm (such as Fair Queue) [57]. The CoDel parameter is adopted in this study to detect the queue sojourn time in

NDN routers.

2.4.2 Scheduling Theory

In computer networks, scheduling is the method by which work specified by some means is assigned to resources that complete the task (job). The task may be virtual computation elements (e.g., share CPU time, threads, and data flows), which are in turn scheduled to hardware resources (e.g., network links, processors, and expansion cards). The main purpose of scheduling theory is to minimize resource starvation and to ensure fairness amongst the parties utilizing these resources. Scheduling deals with the problem of deciding which of the outstanding requests is to be allocated resources. There are many different queue-scheduling disciplines, each one attempting to locate the correct balance between fairness and complexity. Several queue scheduling disciplines are described in [58]: First-in-First-Out queuing (FIFO), Fair Queuing (FQ), Priority Queuing (PQ), Weight Fair Queuing (WFQ), Weighted Round-Robin queuing (WRR), Deficit Round-Robin queuing (DRR) and Deficit Weighted Round-Robin queuing (DWRR). The WRR scheduler is a pioneer in ensuring fairness among different queues. It allocates packets from different flows to separate queues, and removes them from each queue in a cyclic manner in proportion to the weight preassigned to each queue. WRR performs well when all packets have the same size.

The DRR scheduler is an enhancement of WRR to overcome the problem of variable packet sizes [127]. It schedules the packets without knowing the mean packet size of each flow in advance, using bit by bit scheduling. The DRR scheduler gives near-perfect fairness throughput with low implementation cost. The DWRR scheduler is a variation of the DRR and WRR schedulers that handles the fairness between flow based on the weight allocated to each queue. Different queues are allocated a different quantum value using a proportionally weighted function. This study adopts the DWRR scheduler to classify the different flows based on the content name not the IP; the queue heads will be classified into Data queues and Interest queues to differentiate the shaping weight. In addition, the calculation of the quantum of each queue is calculated based on the weight assigned to each flow.

2.5 Summary

This chapter considers the new theoretical shift of Named Data Networking (NDN) by covering in detail many transport control techniques. It first gave an overview of Named Data Network architecture, followed by the transport control component concept (forwarding plane and congestion control) and the challenges demonstrated in this area. Many techniques related to this research problem and scope were explained in detail based on the literature review. A critical reviews of related techniques was achieved in order to highlight and identify gaps in the research. As a result, the proposed techniques aim at mitigating congestion, enhancing link utilization and fairness. In the next chapter, the research framework for attaining the objectives of this research, highlighted in Chapter One, is presented; the following chapters cover the implementation, validation and evaluation of the Explicit Control Agile-based conservative window adaptation (EC-Agile) scheme that adapted scheme called Agile-SD [111] that uses Agility Factor (AF) which reacts quickly to the changing rate between competitive consumers. Next, Shape Deficit Weight Round Robin (SDWRR) scheme that adapted DWRR scheme [128, 127] to create new Interest and Data queues for each flow received by the NDN router and shaped the Interest forwarding rate to utilize the link bandwidth in a fair manner. Also, adopt one of the AQM schemes called CoDel [122] to indicate packet delay in each created queue Finally, the forwarding strategy side implement a Queue-delay Parallel Multipath (QPM) scheme the distribute incoming packets based on the packet queue delay on each paths.

CHAPTER THREE
RESEARCH METHODOLOGY

The general aim of this research is to enhance the scalability and fairness of transport control in Named Data Network (NDN) architecture. To realize this goal, a new mechanism is designed, namely Hybrid Rate Control Mechanism (HRCM), comprising Shaping Deficit Weight Round Robin (SDWRR), Queue-delay Parallel Multipath (QPM), and Agile-based Conservative Window Adaptation (EC-Agile) schemes. The combination of the SDWRR, QPM, and EC-Agile is unique to this research. This chapter introduces the methodology employed in this research to design and implement HRCM, verify and validate the implementation, and evaluate its performance .

This chapter first outlines the overall research methodology framework. Section 3.2 presents the initial plans for the research. Section 3.3 describes the second stage, reviewing previous work. Modelling design is demonstrated in Section 3.4. Section 3.5 presents the experimental design. Verification and validation methods are described in Section 3.6. Section 3.7 proposes the evaluation and analysis of the result. Finally, Section 3.8 summarizes the chapter.

3.1 Design Research Methodology Stages

This study develops a new mechanism for NDN transport control based on queuing, scheduling and congestion avoidance principles, and able to adapt itself to the enhance the behaviour of service transport control. By adapting changes in the behaviour of transport control, HRCM is designed and implemented to produce better performance in link utilization in terms of throughput, download time, delay, queue length and fairness, to meet the requirements of consumers. These requirements match the design research definition as proposed by Blessing and Chakrabarti in [129]. They stated that the research design must be scientific in acquiring valid results in both the theoretical and practical sense. Therefore, the research methodology applied here guides the entire process from beginning to end scientifically and comprehensively. This ensures that the experiments conducted and the results produced are trustworthy, repeatable and comparable.

The specific research methodology consists of several stages: an initial plan for the research, a critical review of previous work, design of the model, testing the system design, verification and validation, evaluation and analysis, research contributions, and reporting the work. Figure 3.1 illustrates these stages and the links between them, with the main process and outcomes for each stage. In the following sections, an explanation of the main concepts and objectives of each stage is provided, with emphasis on the main methods and deliverables at each stage.

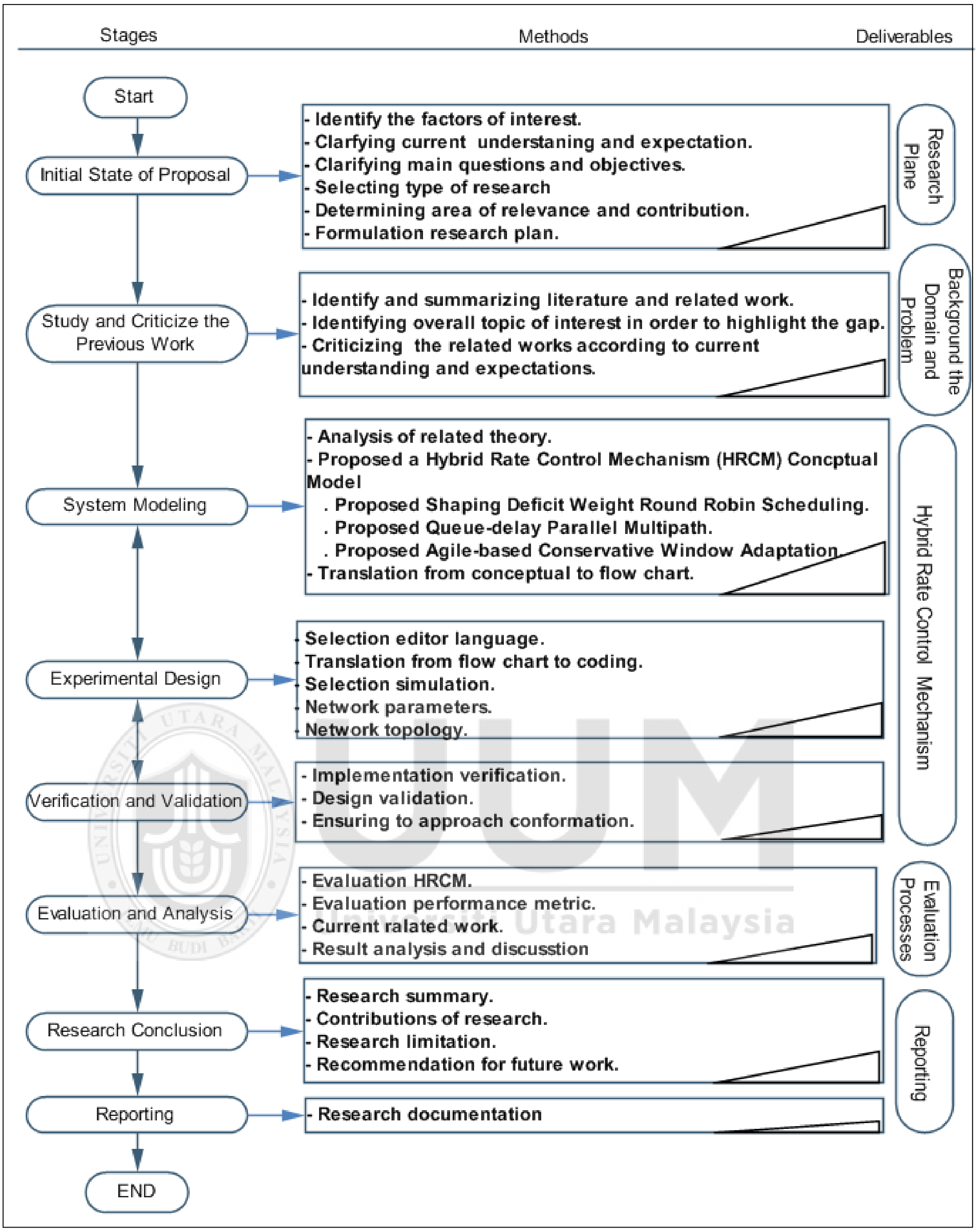


Figure 3.1. Research Methodology Framework

3.2 Initial Research Stage

The first stage in any research is to draw up a research plan in order to highlight the main content of the work and how it is distributed. The initial plan consists of three steps as depicted in Figure 3.2. Step one (research focus) reviewed the literature and highlighted the challenges presented by the NDN paradigm, specifically regarding congestion control and forwarding in NDN architecture, in order to identify the research gap, the objectives, the scope and the research significance. Step two (core thesis contributions) focuses on proposing a Hybrid Rate Control Mechanism (HRCM) that can mitigate congestion, Interest re-transmission and fairness, and enhance network scalability. This mechanism considers the Interest packets rate on the consumer side as well as monitoring and controlling the forwarding rate on the router side to increase fairness between competing flows and prevent link congestion. In this step, SDWRR, QPM, and EC-Agile were designed.

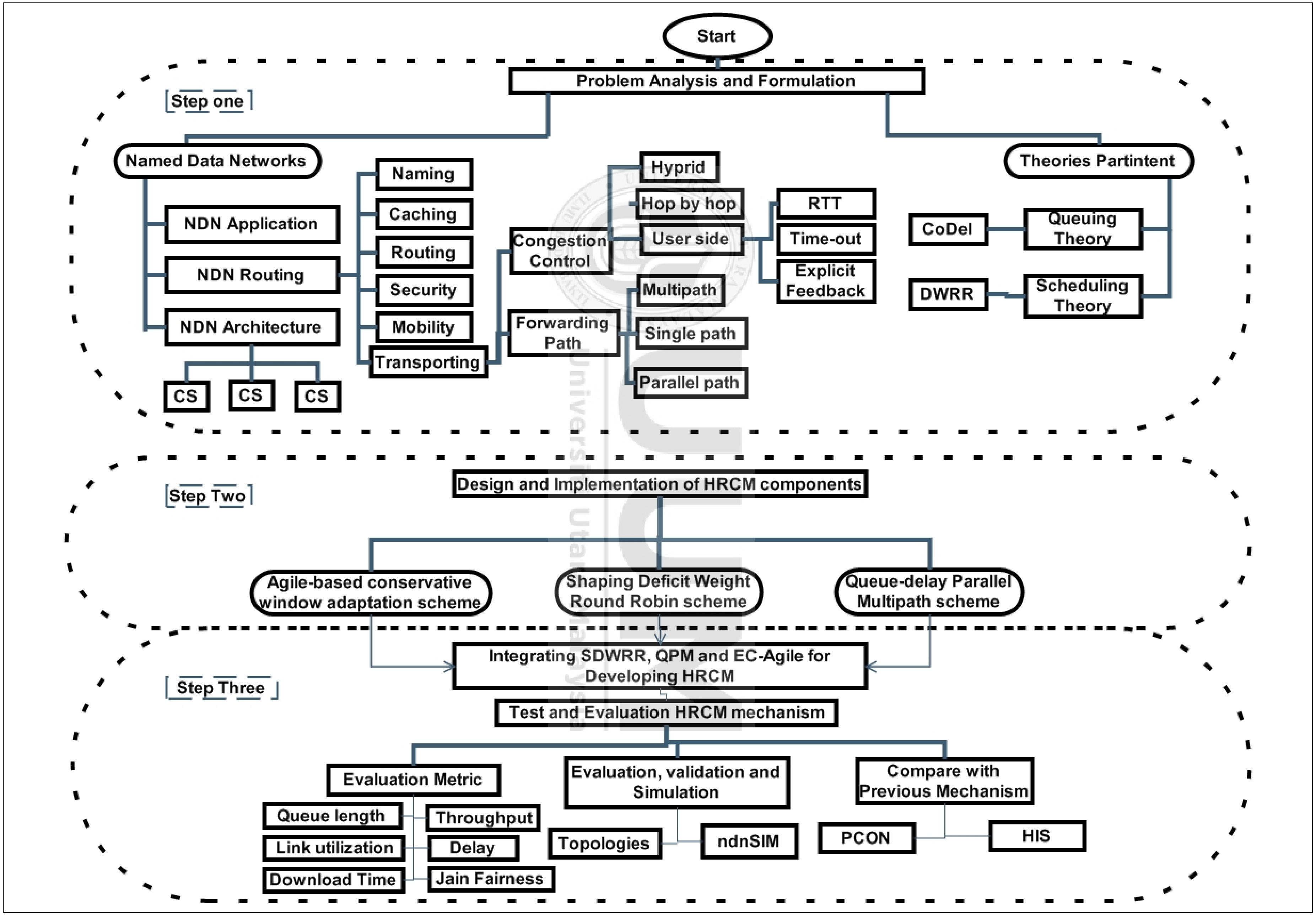


Figure 3.2. Research Plan

Integrating SDWRR, QPM, and EC-Agile into HRCM, testing and evaluating the proposed mechanism are shown in the third step. This step includes the performance metrics, validation and network environment, as well as a comparison with current solutions. It determines the parameters that should be applied, such as link capacity, throughput, fairness in the network depending on the bottleneck topology.

The outcomes of the initial research stage are::

• Forming a research focus and research motivation.

• Forming the research problem, and the research questions in line with the research objectives.

• Identifying the research scope, type of research, research methods.

• Last but not least, stating the expected research contribution and deliverables.

3.3 Overview and Critical Analysis

The second stage of the research framework examined NDN architecture as presented in the literature. This stage was also used to obtain a deep understanding of transport control, in order to reveal the gaps that result in network overload and congestion in NDN routers. Thus, the effect of transport control in the performance of the entire network was identified and highlighted. This stage also criticized the current work addressing the problem of transport control. The researcher identified and discussed the research problem, specifically congestion, fairness and network overload.

Figure 3.3 illustrates the main processes in this stage, each increasing the understanding that leads to refining the conceptual framework of this research.

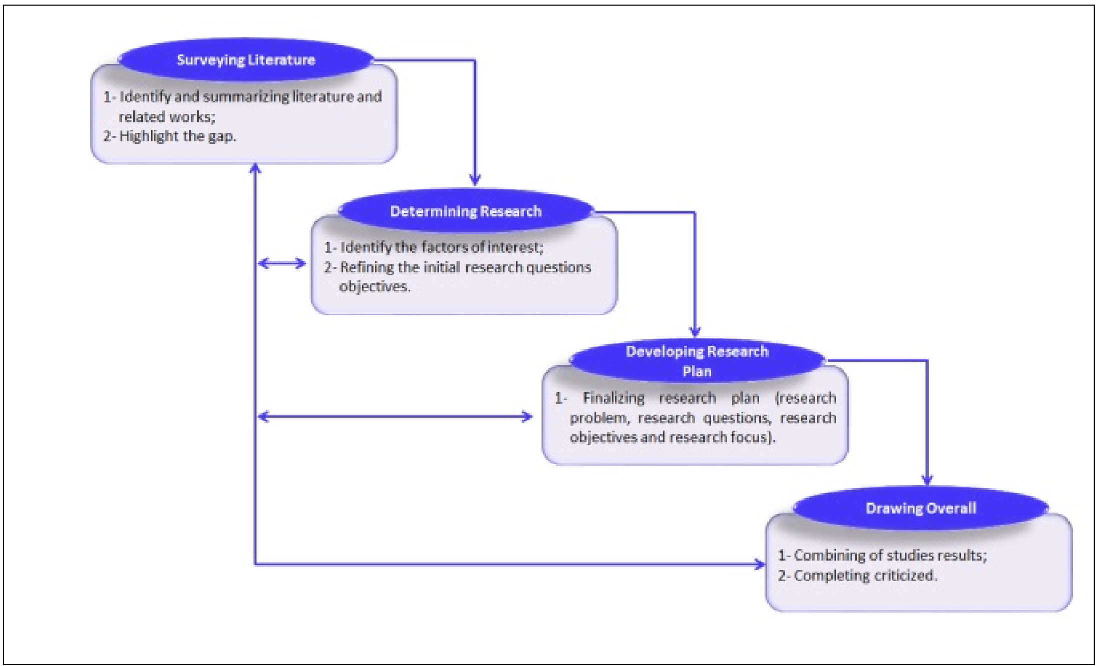


Figure 3.3. Main Steps in Critical Review of Previous Work

The outcomes of the critical analysis stage, as described in Chapter Two, are:

• Identifying the problem of transport control,

• Highlighting the issues in transport control, and

• Critically reviewing the literature related to the current transport control solution.

3.4 Hybrid Rate Control Mechanism Conceptual Model

Conceptual model considered the most important stage in this framework because it presents the proposed mechanism, based on queuing theory, scheduling theory and congestion avoidance to control different flows in the network when links are congested. Queuing theory manages the flow of Interest packets and detects congestion. Scheduling theory controls the dequeuing of packets from different queues in the router interface, and congestion avoidance controls the window size in the consumer node. This will fulfil the aim of this research, to design and implement HRCM in order to mitigate congestion, increase the link utilization and fairness between flows and prevent network collapse.

As transport control consists of congestion control and forwarding functions, HRCM includes these schemes in its design, from the concept, problem identification, through to the objectives and scope. The concept in this research is the three major properties shown in Figure 3.4. It starts from the consumer side by implement an Explicit Control Agile-based conservative window adaptation (EC-Agile) scheme that adapted scheme called Agile-SD [111] that uses Agility Factor (AF) which reacts quickly to the changing rate between competitive consumers. Next, the NDN router interfaces controller implement the Shape Deficit Weight Round Robin (SDWRR) scheme that adapted DWRR scheme [128, 127] to create new Interest and Data queues for each flow received by the NDN router and shaped the Interest forwarding rate to utilize the link bandwidth in a fair manner. Also, adopt one of the AQM schemes called CoDel [122] to indicate packet delay in each created queue Finally, the forwarding strategy side implement a Queue-delay Parallel Multipath (QPM) scheme the distribute incoming packets based on the packet queue delay on each path . These schemes are combined in the single transport control mechanism called HRCM, in order to monitor, detect and manage the forwarding in NDN, to achieved the research goals presented in Chapter One.

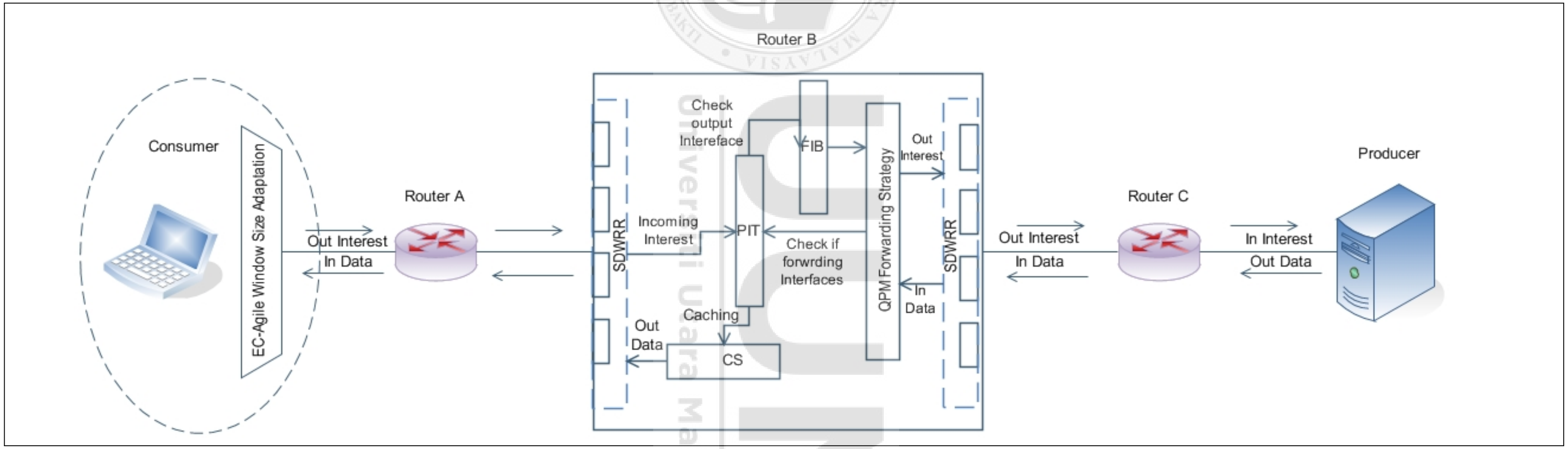


Figure 3.4. HRCM mechanism

Figure 3.4 shows each mechanism in detail with the order and position of each function. From the left of Figure 3.4, EC-Agile has three functions: sending Interest packets, and receiving Data and NACK packets. On the router side, both SDWRR and QPM have several functions, discussed below.

The SDWRR responsible about the enqueue, dequeue and the scheduling processes in NDN router as shown in Figure 3.4. Furthermore, SDWRR was designed to control and shape the incoming Interest packets by calculate weight that use the sojourn time of both Interest and Data queue packets and explicit notification if any to control the Interest forwarding and detect congestion in NDN router. Also, SDWRR create a NACK and sent it to downlink as congestion explicit notification if the sojourn time of Data queue of certain prefix exceed the threshold.

QPM was designed in the NDN router to manage and distribute the incoming Interest packets to all available path in a parallel manner. QPM send a signal to SDWRR in each available path listed in FIB for the coming prefix to create a Interest queue if it new prefix and divide the Interest packets equally between interfaces. After that, QPM uses sojourn time of Interest queue as an indicator to control the distribution of the next incoming Interest to each path in the network by given a weight to each available path using the forwarding control module as shown in Figure 3.4. If all Interest queues of the available paths become congested QPM mark the Data packets and sent it downlink as congestion explicit notification to slow down their sending rate.

EC-Agile was designed for the consumer to adapt and manage the Interest window rate. It starts by forwarding the Interest packet based on the slow start as each time the consumer receives a Data packet, the send rate will increase one window size. However, when a consumer receives marked data, NACK or packet time out congestion avoidance in EC-Agile starts working. The increasing fraction in EC-Agile congestion avoidance is calculated based on the gaps between *cwnd* and the last window from which the consumer received the last notification to reach the stability faster. More detail about the implementing schemes will be discusses in chapter four of this thesis.

3.5 Design of Experiments

In implementing any model it is advisable to adopt a reliable, verified and scholarly approved simulation program, as is the case with HRCM. The simulation for generating traffic Interest was used for verification and validation purposes. This section focuses on editor languages, common simulators of the ICN and/or NDN architectures, simulation settings, and networking typologies, based on the literature.

3.5.1 Editor Language

The proposed mechanism was transformed into C++ code, the base programming language of ndnSIM. The Eclipse C/C++ Development Tool (CDT) [130], running on top of the Eclipse platform, was used verify to that the model had been coded properly and was free of bugs or errors. The Eclipse IDE for C/C++ Developers provides advanced functionality for developers, including an editor, debugger, launcher, parser, and file generator, as shown in Figure 3.5.

CDT's Code Analysis (CodAn) integrated in Eclipse can assist the researcher by indicating possible syntax errors as he or she types in the code, finding bugs and other potential problems. It works by scanning the C++ code and checking for potential programming problems as well as syntax and semantic errors, as shown in Figure 3.5.

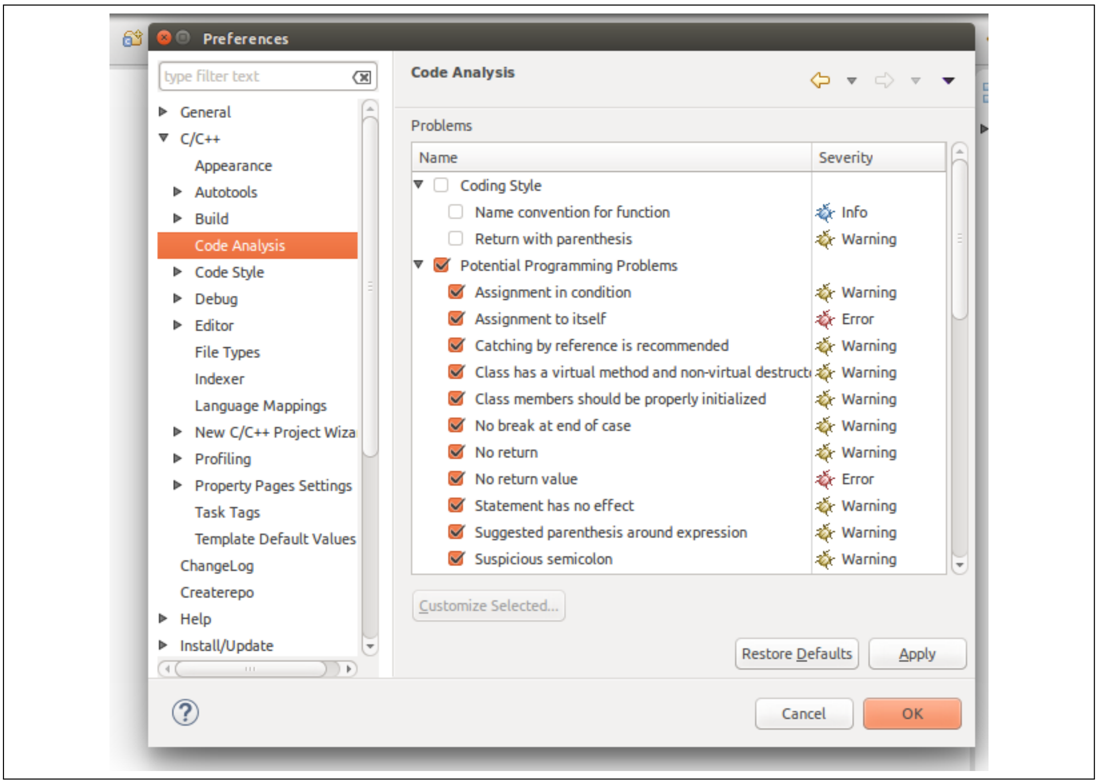


Figure 3.5. Eclipse CDT's Code Analysis

3.5.2 Simulators Selection

Simulation tools are widely used in dynamic scenarios, especially networks and real systems. The simulator is a computer-based system model or is generated using computer programming. Simulation is a more flexible tool for studying the performance of various protocols [131]. and was therefore the chosen method for performance evaluation in this study, representing the dynamic behaviour and responses of real systems. Many discrete-event network simulators are available, including both commercial products for purchase and open-source products that can be downloaded and modified. Some of the most popular tools used by NDN researchers are OPNET [132], OMNET++ [133], and Network Simulator 3 [134].

The first simulation build for Content Centric Network (CCN) a simulation called ccnSim [135]. CCNSim is a chunk-level simulator for the CCN architecture, designed in C++ under the framework of OMNET++. The principal advantage of ccnSim is

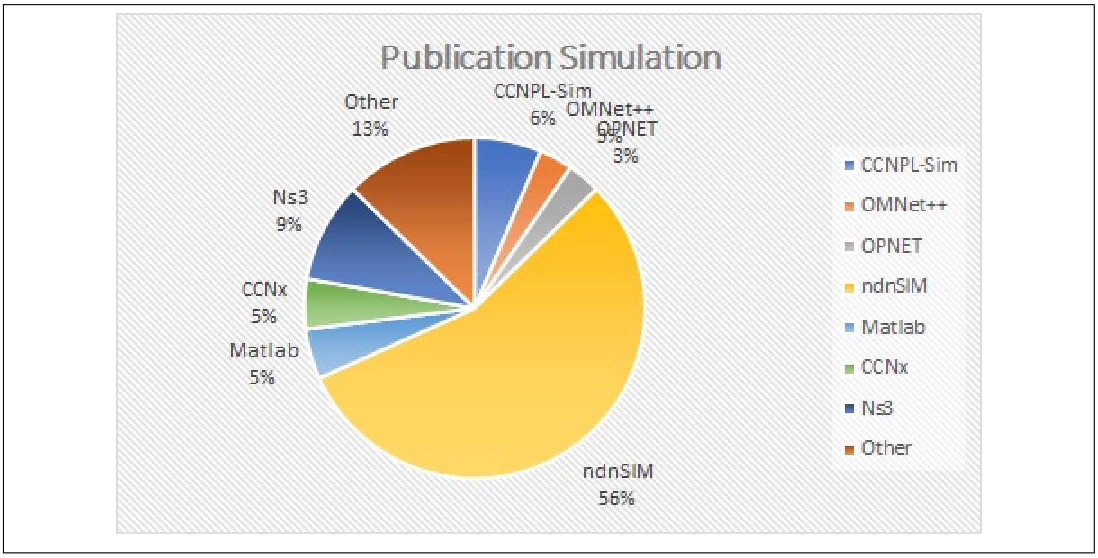


Figure 3.6. Percentage of Each Simulation Tools used in Literature

its versatility, permitting addressing situations with expansive CCN cache size (up to 10674 chunks) and catalogue sizes (up to 108 contents) on off-the-shelf item equipment [136]. According to Afanasyev in [137], ccnSim was composed and executed with the primary objective of executing experimentation of various cache replacement policies for the NDN router. As such, it cannot be considered a fully showcased execution of the current NDN design. In the present adaptation of ccnSim, the FIB and the PIT segments are actualized in the least complex conceivable way. This implies that it is incapable of assessing the diverse information-sending procedures, distinctive directing arrangements, or diverse congestion control techniques.

The other simulation that build for CCN/NDN built in NS3 framework. As NS3 is a discrete-event simulator which can be used for the implementation of numerous applications. The NS3 project [138] is a free, open-source network simulator available for teaching, the research community, students, and development work under the General Public Licence, version 2 (GPLv2). NS3 has numerous external animators and tools. This simulation platform provides users with a single, integrated Graphical User Interface (GUI) environment, data analysis, and visualization. NS3 has been designed in modular fashion as a set of libraries which may be combined together as well as with other external software libraries. Currently, NS3 can be installed only on the Linux operating system in native mode. The non-availability of backward compatibility with NS2 also hinders the ready acceptance of NS3 as the default simulation tool since NS2 users will be reluctant to abandon it immediately.

According to Pentikousis et al. in [139], simulators and emulators must be able to capture faithfully all features and operations of the respective information-centric architectures. NDN architecture needs flexible simulators to support ease of use, usability, configurability, simplicity, and logical programming dynamism to simulate NDN. NDN requires that simulators are easily extendable through open-source options. Adequate documentation and manual guides are needed to easily handle simple and complex network experiments through increasing network sizes, nodes, time, parameters, and metrics selection. The following simulation models for CCN/NDN architecture have been built in NS3:

- CCNPL-Sim is written in C++ [140]and is based on CBCBsim, from which it imports part of the forwarding layer and the Combined Broadcast and Content-Based (CBCB) routing protocol, although the features of CCN protocol have been designed from scratch. The simulator has been used to evaluate per-hop sending behaviour and publisher-based congestion control, where a fine-grained control over individual packets is basic to get precise execution results. CCNPL-Sim is the main CCN/NDN simulator to offer out-of-the-box implementation of flow control algorithms, like Additive Increase Multiplicative Decrease (AIMD), thus representing a characteristic decision to maintain a strategic distance from the weight of a comparable usage from scratch.CCNPL-Sim has the drawback of utilizing a custom discrete-event simulator. Thus, aside from the few scale studies that concentrated on congestion control, other simulators might be preferable for more extensive purposes or vast-scale research; for example, ccnSim is a superior fit [136].

- ndnSIM module is part of the NDN tool set which permits the execution of NS3 simulations (as discussed in [137, 141]).The development of ndnSIM has the following objectives [141]:
  - It is an open-source bundle able to execute the activities on a typical simulation framework.
  - It reliably simulates the entire operations necessary for the NDN protocol.
  - It maintains packet-level compatibility with CCNx execution, to permit sharing of the activity estimation and packet examination tools between CCNx and ndnSIM. It coordinates use of real CCNx traffic traces to drive ndnSIM simulation tests.
  - It is ready to bolster expansive-scale simulation tests.
  - It facilitates network-layer experimentation with directing, content caching, packet sending, and congestion administration.

Following the NDN design, ndnSIM is actualized as another model of the network layer protocol that may keep executing on top of any accessible model related to available link-layer protocols (e.g., wireless, CSMA, and point to point) as well as on top of a network layer (e.g., 1Pv4 and 1Pv6) and transport layer (e.g., TCP and UDP) protocols. This adaptability enables it to reproduce situations of different homogeneous and heterogeneous sending scenarios (e.g., NDN-only and NDN-over-IP). The simulator is actualized in a particular mould, utilizing separate C++ classes to model the behaviour of every network layer element in NDN, This modular structure permits any segment to be effectively adjusted or supplanted with no or insignificant effect on different parts. In spite of the central protocol stack, ndnSIM incorporates various

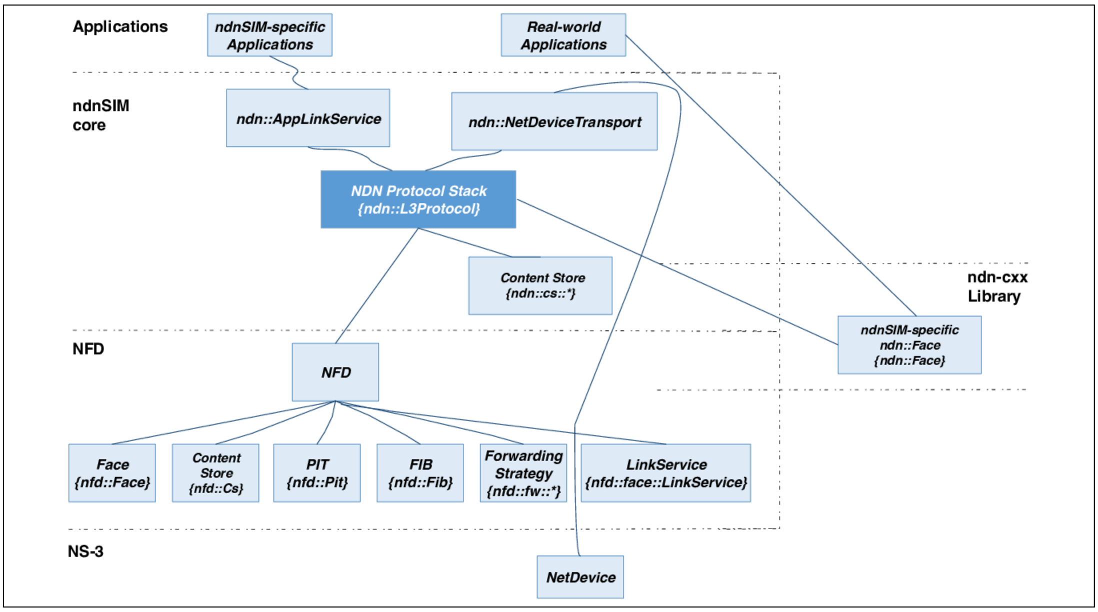


Figure 3.7. ndnSIM Structure

fundamental activity generator applications and aids classes to rearrange formation of simulation scenarios (e.g., helper to install NDN stack and applications on nodes) and tools to accumulate simulation insights for estimation purposes.

Figure 3.7 demonstrates the essential collaborations among L3 Protocol, Face, Content Store,PIT, FIB and Forwarding Strategy in ndnSIM. Each element in line with the core exception L3 Protocol has a great amount of substitute implementations which might be randomly selected by the simulation scenario with the help of classes, known as helper classes (for more detail see http://ndnsim.net/helpers.html).

Many simulators are obtainable for CCN/NDN with similar schemes. Figure 3.6 reports the results of the studies mentioned in chapter 2 claim to utilize their custom simulations and papers. Generic tools, such as NS3, Matlab, Omnet++, and QualNet, are mentioned with no indication of the alterations required and/or with no reference to the code used. The dominant part in CCN/NDN simulation and the most popular simulator is ndnSIM. On the other hand, about two-thirds of the results are not reproducible, either because the researchers have not specified the tool used for the

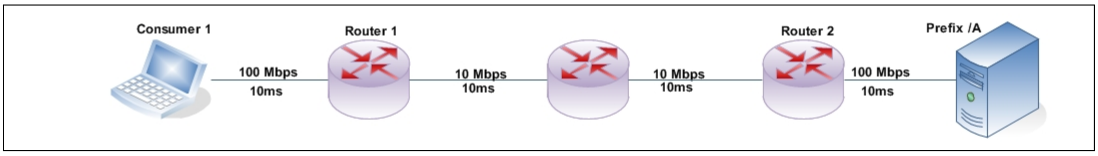


Figure 3.8. Baseline Topology

evaluation part of their proposal, or because they have used a custom simulator. Table 3.1 compares the following three simulators: CCNPLSim, ccnSim and ndnSim.

Table 3.1
Evaluation Between Different Simulations

Criteria	ccnSim	CCNPL-Sim	ndnSIM
Real code execution	N	N	N
Debugger support	Y	Y	Y
Tracing support	Y	Y	Y
Scalability	High	N	High
Deployment	Moderate	Moderate	Moderate

3.5.3 Topology Selection

According to [139], "there is no single topology that can be used to easily evaluate all aspects of the ICN paradigm". In this research, several network topologies with different network sizes and varying numbers of nodes were used to validate and evaluate the proposed model. More specifically, the scenarios are applied in Baseline, Dumbbell and Abilene topologies [142]. The experimental outcome for every simulation scenario compares the simulation results from related work with the model's simulation results.

Baseline topology is a common topology used in many network simulations [58]. Several researches [38, 143, 144, 145] have used it for evaluating the window size, queue overflow and download time in CCN/NDN. It has been used to study the impact of competing flow rates, link utilization and fairness between consumers and publishers. A simple Baseline topology (see Figure 3.8) containing five nodes (i.e., one consumer

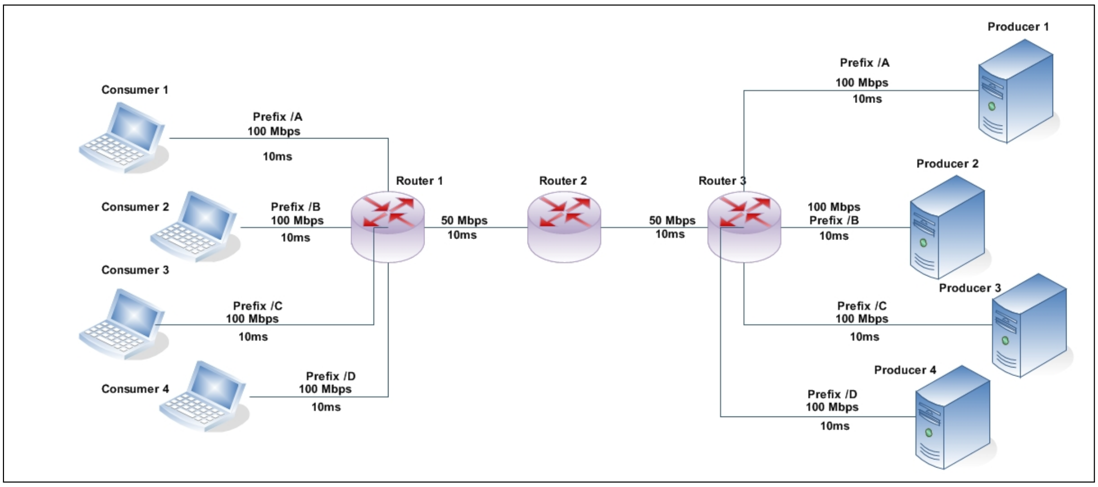


Figure 3.9. Dumbbell Topology

node, one producer node, and three NDN routers) and four links.

Dumbbell topology has been widely used in congestion network simulations [58] and several researchers [38, 42, 96, 145, 144, 143] have used it for performance evaluation of congestion control in CCN/NDN. It has been used to study the impact of competing for flow rate, link utilization and fairness between consumers and publishers. A simple Dumbbell topology of eleven nodes (i.e., four consumer nodes, four producer nodes, and three NDN routers) and ten links, is illustrated in Figure 3.9.

The third topology is based on the Abilene network, created by the Internet2 community and connecting regional network aggregation points to provide advanced network capabilities to over 230 Internet2 university, corporate, and affiliate member institutions in the US [142, 143]. Recent studies have recognized the importance of the Abilene topology with emphasis on the probing result introduced. Abilene topology in this study (see Figure 3.10) consists of eleven NDN routers, four consumer nodes connected to Atlanta node, eight producer nodes four connected to Sunnyvale node and the other four connected to Seattle node that give a total of 26 links in general.

Figure 3.10. Abilene Topology

3.5.4 Simulation Settings

All the experiments presented in Chapters 4 and 5 were performed using ndnSIM. A comprehensive simulation measuring the many performance metrics is presented in Section 3.7.1; it used a machine with the Linux Ubuntu 14.04 operating system, because ndnSIM works most efficiently in the Linux environment. The hardware is Intel Core (TM) i7-3612QM at 2.10 GHz CPU, 8 GB of DDR3 RAM.

The full range of default parameters that might have an impact on the experiment mentioned in Table 3.2 and other parameter changed every time to strengthen the validity of the simulation results. Network topologies with different network sizes were used to test and evaluate the proposed model, using the Dumbbell, Abilene and Baseline topologies. The experimental outcome for each HRCM simulation scenario is compared with the simulation results of related work.

Table 3.2
Design Range for the Simulation Parameters

Parameter	Description
Simulation Environment	ndnSIM
Network Type	Local Network
Link-layer Protocol	PointToPoint
Simulation Topology	Baseline, Dumbbell and Abilene
Content Store	10000 Data packets
Content Size	1GB
Queue Length	1000 packets
Data Packet Size	1024 byte
CS Replacement Policy	LRU
Simulation Time	100s

3.6 Verification and Validation

Model verification evaluates the integrity of the transformed model that was illustrated through flowchart or pseudo code to an executable computer program [146]. The simulator used in this research was built based on C++. Therefore, in this research, the structure of the proposed schemes will be implemented using C++ as a programming language. In addition, all schemes must be verified to assure that code was written correctly without errors or bugs [147, 148]. Eclipse Integrated Development Environment (IDE) [130] is implemented for this purpose. Many functionalities are provided by the eclipse IDE for C++ developers such as, editing, debugging, launching, parsing as well as generating for creating files. Eclipse IDE supports C++, Java, PHP, C, and HTML5.

In order to verify implemented schemes, Eclipse C/C++ Development Tool (CDT) software is used [130]. The Eclipse C/C++ Development Tool (CDT), running on top of the Eclipse platform, was used verify to that the model had been coded properly and was free of bugs or errors. The Eclipse IDE for C/C++ Developers provides advanced functionality for developers, including an editor, debugger, launcher, parser, and file generator. CDT's Code Analysis (CodAn) integrated in Eclipse can assist the researcher by indicating possible syntax errors as he or she types in the code, finding bugs and other potential problems. It works by scanning the C++ code and checking for potential programming problems as well as syntax and semantic errors, as shown in Figure 3.5.

Balci in [149, 146, 150] defined validation as the affirmation of the implemented model, policy or a mechanism that acts accurately compared to other validated models. The accurate ratio has to be acceptable and the behaviour has to be consistent with models and simulations. Hence, the validation perspective is about to build a right model. The right model means the introduced mechanism performs the basics and required functions properly.

Validation techniques were discussed in [146, 150]. These methods are informal, formal, static and dynamic. Each main method has sub-techniques. The informal method depends on the human factor; neither rigorous mathematical rules nor guidelines exist in this method. Informal method is usually applied in robust approaches based on formal guidelines. Audit technique [151] is one example of informal method. The formal method is based on mathematical proof. If the mathematical model of proof is correct, that means the model is valid. However, not all models are obtainable to prove mathematically their correctness. In addition, based on the current state of the art, many formal techniques are not applicable for complex and reasonable simulation models. Inductive and inductive assertions are examples of formal method techniques [151, 152]. The static method concerns about the truthfulness of model's aspect. This method does not need an actual implementation of the model. Rational implementation is sufficient to validate the model. The static mode has information about the flow structure of the model, the source code and data. This method can be implemented for automated tools. The simulation compiler represents [151] and Cause-Effect Graphing are examples of this method [153, 154].

Dynamic method requires an implementation for the model. The validation is based on the behaviour of the implemented model. The dynamic method is implemented through three phases. The first phase is adding a source code into the body of the executable model. The inserted codes collect information about the implemented model. Then, the model is executed. Finally, the output of the executed model will be compared with other valid models. The verdict of the validation will be based on the behaviour of the output model compared to the valid models [150, 155]. Alpha testing [156] technique is an example of this method. Since this study is focused on Forwarding and congestion in dynamic environment, the dynamic method will be implemented to validate the three schemes SDWRR, QPM and EC-Agile separately . The output of each scheme will be compared with the output of other validated mechanism using the same simulation tool based on the same experiment setup and environment. The validation will be implemented by simulating the schemes for numerous times using different numbers of jobs to measure the sensitivity of the schemes [157, 158]. The results that are generated from simulating schemes will be presented graphically. Then the behaviour of the schemes will be compared with valid schemes [153] based on the graphical lines [159, 160]. The validation of schemes touches three techniques in dynamic approach, which are Sensitivity Analysis [157, 158], Graphical Comparisons [159, 160] and Comparison Testing [153].

3.7 Performance Evaluation

Performance evaluation is a crucial step in evaluating the final results of any research project [155, 161]. It is required if a system designer intends to compare a number of alternative designs for finding the accurate design [162, 163, 164]. Accordingly, different NDN architectures were evaluated in the literature review, using a mixture of theoretical analysis, empirical measurements (testbed) and simulation or emulation techniques. Researchers usually follow a specific methodology based on the objectives of their experiments (e.g., to evaluate scalability, quantify resource utilization, or analyze economic incentives).

In addition, the experimental process itself in addition to the evaluation methodology are now actively investigated in NDN architectures. There are many factors that can affect the experimental results, such as network condition (e.g., Available link capacity, topology selected, link delay, node mobility, background traffic load, loss rate characteristics, disruption patterns, and the variety of devices used) [139, 26]. This research evaluated HRCM by comparing it with the current work, after integrating the three schemes SDWRR, QPM and EC-Agile.

3.7.1 Evaluation Metrics

The key step in performance evaluation is the selection of performance metrics [163, 162, 165]. Therefore, [155, 131, 164] argues that performance metrics can mean different things to different researchers depending on the context in which they are used. Thus, their selection is important in investigating the behaviour of the mechanism from different viewpoints. The use of multiple different metrics gives a complete picture of the performance of the proposed mechanism.

There are four criteria for choosing suitable metrics for performance evaluation in simulation techniques: they should be readily available or simple to implement in ndnSIM; they should be either the most recent or the most famous; it is preferable if the mechanism is utilized in real-life routers; and finally, previous studies should have used them. This study focuses on link utilization, throughput, download time, delay, Interest rate, queue length and fairness as the metrics used to measure performance;

these metrics have been used by other researchers in previous studies.

- Throughput: the total number of packets received by the consumer per unit of time (i.e., experiment time per seconds). In other words, throughput is measured as a flow-based metric of per-connection transfer times. An efficient congestion control mechanism results in a significant increase in throughput which is subject to the demand on the application, and to environmental constraints [40]. The following formula is often used to calculate the throughput value:

$$TH = \frac{TN}{T} \tag{3.1}$$

Where

$TH = Throughput$

$TN = The\,total\,number\,of\,Data\,packet\,received$

$T = Time\,interval$

- Download Time: the total time required for the consumer to download one data file [60].

$$DT = \frac{FS}{DS} \tag{3.2}$$

Where

$DT = Download\,Time$

$FS = File\,Size$

$DS = Download\,Speed$

- Link utilization: the total throughput over transmissions link [82].

$$LU = \sum_{n=1}^{n} TH \tag{3.3}$$

Where

$LU = Link\,Utilization$

$TH = Throughput$

- Queue length: the average of the packet occupied by the aggregation of the flows. It is an important metric to measure queue stability and can be calculated by:

$$ql = \frac{1}{N}\sum_{i=1}^{n} q_i \tag{3.4}$$

Where

$ql = Queue\,Length$

$N = Number\,of\,Flow$

$q_i = Queue\,occupation\,by\,flow\,i$

- Delay: the total time required to send an Interest packet from the receiver node to the destination node and receive the data packet from destination. This is often referred to it as Round Trip Time (RTT) [40].

$$RTT = T_r - T_s \tag{3.5}$$

Where

$T_r = Data\,received\,time$

$T_s = Interest\,sent\,time$

- Jain Fairness the mathematical formulation to measure the fairness among number of links by calculating the received throughput for each link, it is originated by Raj Jain in 1984 [166] and formulated as:

$$J(x_1, x_2, \ldots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n * \left(\sum_{i=1}^{n} x_i^2\right)} \tag{3.6}$$

Where

$J = Fairness\,index$

$x_i : Throughput\, for the\, link\, i$

$n : Number\, of\, link$

3.7.2 Results Analysis and Discussion

There are two sources of results. First, results are accumulated from the simulation or implementation of the proposed scheme and comparison with related work (see Chapter Four). Secondly, they are collected from the evaluation of HRCM and comparison with previous work (see Chapter Five). Both results necessitate multiple analysis which may be different or similar to each other in some aspects. The former experiments which are related to the implementation of the proposed scheme require specific data analysis related to performance, although data analysis for evaluation of the mechanism takes a different form.

3.8 Summary

This chapter has detailed the methodology and research design used to achieve the research objectives. A specific methodology was used to realize all the research steps, starting from understanding the problem through to designing and implementing the HRCM components in order to solve the problems of link congestion, network overload and fairness. The experimental design and methods of verification, validation, and evaluation were also described in this chapter.

CHAPTER FOUR
HRCM DESIGN AND IMPLEMENTATION

Chapters One and Two thoroughly illustrated the background of this research through the introduction and the literature review. Chapter Three established the research methodology as a guideline to achieve the research objectives, and illustrated all the steps needed for performance evaluation of HRCM. This chapter designs and implements a the schemes of Hybrid Rate Control Mechanism (HRCM), that aims to enhance the NDN transport control by controlling, monitoring and scheduling the transmission flow, in order to mitigate network congestion and enhance forwarding rate, link utilization and fairness. Section 4.1 Introduction about proposed schemes and the notation used in the implementation. Section 4.2 explains Shaping Deficit Weight Round Robin in detail (theory, description, model analysis, verification, validation, and evaluation). Queue-delay Parallel Multipath scheme and Explicit Control Agile-based conservative window adaptation scheme are covered in detail (theory, description, model analysis, verification, validation, and evaluation) in Sections 4.3, 4.4 respectively. Finally, Section 4.5 concludes the chapter with a summary.

4.1 Introduction

The specific problem considered in this research is to design and implement a controlling and monitoring transport mechanism to avoid congestion in the NDN. This is realized in HRCM, a new transport control mechanism on NDN to monitor, shape and control each incoming flow, indicate congestion and notify the consumer, improving link utilization, stability and fairness. HRCM's three schemes are described comprehensively in this chapter.

HRCM includes all transport control features and concepts (forwarding plane and congestion control; see Section 2.1), each scheme within it designed to accomplish the objectives presented in Chapter One. In the following sections, design, procedure and functions of the schemes are detailed. The notation used to simplify the explanation of the transport control schemes are defined as follow:

Consumers: Let us consider $\mathscr{U} = \{1,...,u\}$ as a set of consumers; each consumer node $u \varepsilon \mathscr{U}$ includes Agile-based conservative window adaptation mechanism to produce and manage Interest packets. The consumer's node starts to produce Interest packets for chunk i and sends it to the network with an initial window size w_u.

Router: We denote $\mathscr{R} = \{1,...,r\}$ as a set of all routers available in the transmission path between consumer and producer server, where each router $r \varepsilon \mathscr{R}$ is connected to some other node(s) via intermediate link(s). We denote $\mathscr{L}_{\mathrm{r}} = [1_r,....,l_r\}$ as a set of links associated to node r, where each link has capacity C_l, for $l_r \varepsilon \mathscr{L}_{\mathrm{r}}$. Furthermore, we denote $\mathscr{F}_{\mathrm{r}} = \{1_r,...,f_r\}$as the set of the interfaces associated to router r, connecting r to other neighboring node(s) through the use of intermediate link(s) $l_r \varepsilon \mathscr{L}_{\mathrm{r}}$.

Router Queues: We consider that each router r has a queue q in each interface $f$as $f \varepsilon \mathscr{F}_{\mathrm{r}}$ denote as q_f and each interface in router associate with Deficit Round Robin (DRR) network scheduler that allocate queue to each outgoing prefix $\mathscr{P} = \{1,....,p\}$denote as q_{f_p}. Each queue associated with queuing scheme to calculate the queuing sojourn time ST of each packet in the queue and each queue ST should not exceed certain time threshold τ.

Flow: We use $X = \{1,.....,x\}$ to denote the flow rate of Interest packet I and corresponding Data Packet D as pairs, where flows are distinguished by looking to prefix names that are common to all Interests/Data of the same object. The incoming packets are classified into flows, and queued to the outgoing queue based on their prefix names as Interests/Data of the same flow will be queued to different outgoing faces. In addition, $x_{p_i}(t)$ donate it as incoming Interest rate of prefix p in the router r.

4.2 Shaped Deficit Weighted Round Robin (SDWRR)

NDN routers need to define efficient mechanisms for traffic control when numerous consumers compete to access the same or different resources, which may lead to link congestion. As a result, this affects link utilization and fairness. Therefore, in network architectures that usually have limited resources, traffic shaping is performed using scheduling to share the available bandwidth among the different applications using the network. Scheduling is also performed to overcome the unfairness caused by possibly different packet sizes used by different flows. Several studies have suggested combining the scheduling and queuing mechanisms to improve fairness and link utilization. One of these is the Deficit Weighted Round Robin (DWRR) scheduler proposed by Shreedhar and Varghese [128], which has low complexity but allows fair and weighted sharing of limited resources at the same time. We adapted the DWRR scheme to create new Interest and Data queues for each flow received by the NDN router and shaped the Interest forwarding rate to utilize the link bandwidth in a fair manner.

4.2.1 SDWRR Design

The SDWRR scheduler adapts DWRR to build two queues for each flow, one for Data packets and the other for Interest packets, and classifies them by content name, not the IP, as demonstrated in Figure 4.1. The queue's head of each flow will be classified to differentiate the shaping weight of Data queues and Interest queues.

After SDWRR has created the two queues for each flow, stable queuing and dequeuing

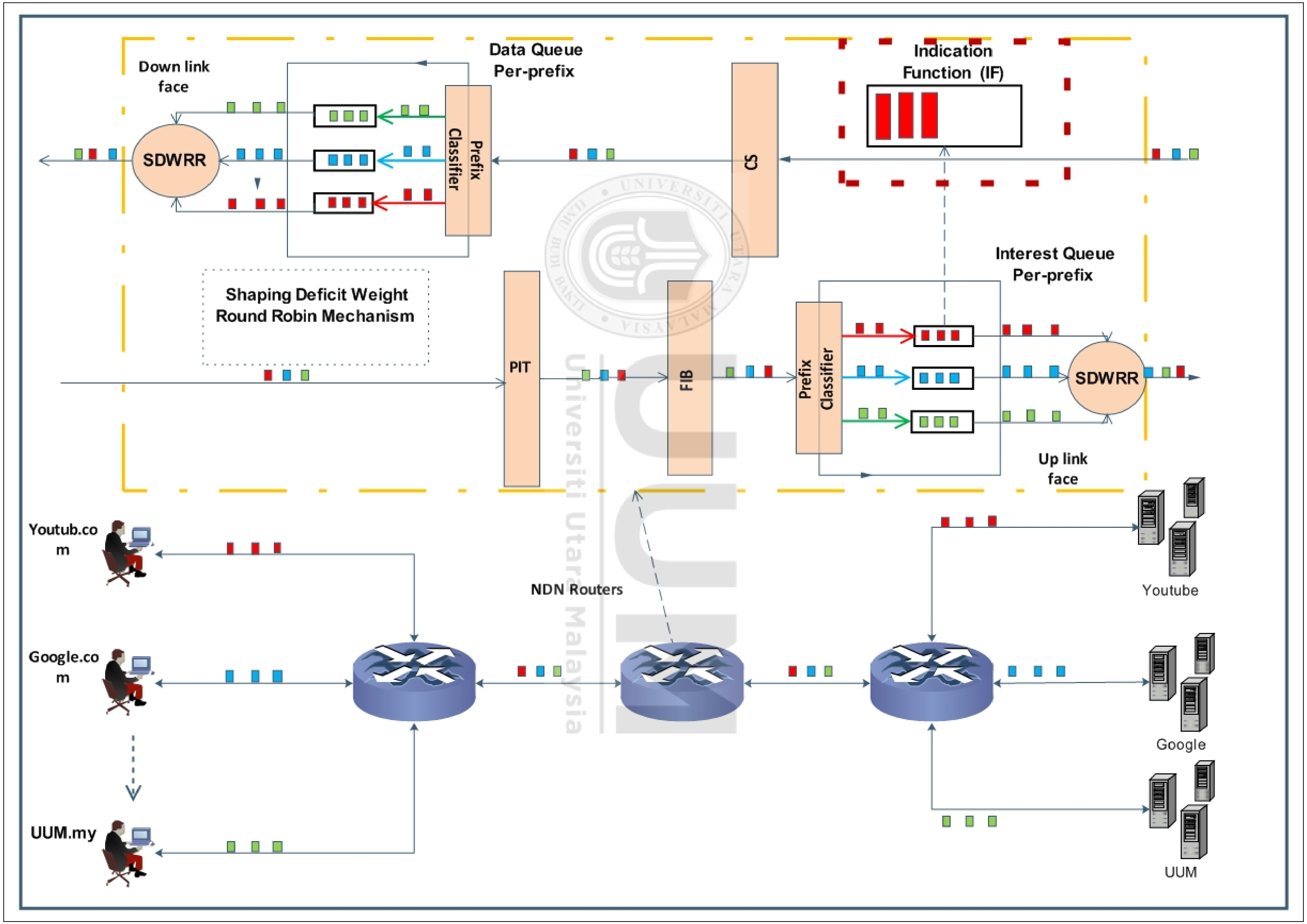


Figure 4.1. SDWRR Model

functions are required. The queuing functions combine an Indicator Function (IF) and control Function (CF). The Indicator Function (IF) is responsible for indicating potential congestion before it happens. Many parameters are used to make this prediction, such as queue length, arrival rate, packet loss and any combination of these. The tuning of these parameters in SDWRR schemes is very important as they can affect the stability of the scheme and the speed of indication to reflect on the overall network. The most popular types of IF described in the literature are rate-based, queue-based and delay-based. The main goal of the rate-based congestion indicator mechanism is to maintain the arrival rate at the buffer at the target value, by controlling the arrival rate affecting the number of packets that can enter the buffer and therefore indirectly controlling the queue length.

The queue-based congestion indicator mechanism controls the queue length by maintaining the packets in the buffer at the target value (threshold); this value is still non-zero. Therefore, in some cases, the targeted queue length can be configured in the queue-based congestion indicator. This value is suitable for some specific network scenarios but may be inappropriate for others. How close the target is to the actual situation is highly load-dependent. Additionally, this non-zero target value can cause a non-zero queuing delay. Therefore, using queuing delay instead of queue length can be a more meaningful measure for end node application. Furthermore, queuing delay by itself is independent of the link capacity and can be used to achieve high link utilization.

As queue delay (packet sojourn time) is a more meaningful indicator than queue length, the source can adjust the delay based on the queue delay (for example the RTT calculation in TCP). Thus, the transmission protocol considers a perfect measure parameter in calculating its delay time. In addition, using the queue delay in IF avoids the effect of different RTT flows so that all flows will be treated alike. Therefore, equalizing the queue delay of different RTT flows will give an equal share of the output link capacity.

IF is defined as a key parameter that should be included, both to avoid congestion and to achieve fairness. SDWRR with CF converges to accomplish efficient buffer management. The main objective of CF is to calculate the weight of each flow and take action to drop or mark the packets that do not satisfy the condition of IF and to improve the stability of the network. As mentioned above, CF will check the sojourn time of each queue; if one of the queue sojourn times exceeds the threshold the scheme will stop sending packets to that queue and sends a notification signal downstream to slow down the sending rate. Otherwise, the sojourn time of each queue will be calculated to assign its weight and use it to calculate the shaping rate. SDWRR divides the queue into Interest and Data queues. CF calculates the weight of each flow as follows:

i. Interest queue weight: CF checks the sojourn time of the Interest queue, and if it is above the target it marks the incoming Data packet of the same prefix and sends it downlink. If it is not above the target CF will check the number of the Data queue of the same prefix and take the lowest sojourn time to calculate the weight. When the sojourn time of the Data queue downlink increases, the Interest rate will decrease; if it decreases the Interest rate will increase.

ii. Data queue weight: CF checks the sojourn time of the Data queue and if it is above the target sends a NACK packet downlink. If it is not above the target CF will assign the weight equal to one.

The next sub-section describes the main procedures combined in the SDWRR scheme: the enqueue procedure in 4.2.1.1 and dequeue procedure in 4.2.1.2.

Algorithm 4.1 Shaping Deficit Weighted Round robin (*Enqueue Function)*

```
for(i = 0; i < r; i = i + 1)
DC_i = 0;
on arrival of packet p
i = ExtractFlow(p);
if(ExistsInActiveList(i) == FALSE) then
  if(i = Interestpacket(p))
    InsertActiveList(i); //*Interest-queue active list*//
     DeficitCounter_i = 0;
      w_i = 0
     Attachtimestampinpacket Header
     Enqueue(i, p); //* enqueue packet p to queue i*//
  else
     InsertActiveList(i); //*Data-queue active list*//
     DeficitCounter_i = 0;
     w_i = 0;
    Attachtimestampinpacket Header
     Enqueue(i, p);
else
  if(queueSize < queueLimit)
     Attachtimestampinpacket Header
     Enqueue(i, p);
  else
     Drop(p);
```


4.2.1.1 Enqueue procedure

As mentioned above, SDWRR has two stages in forwarding each received packet, and this sub-section details the procedure of the NDN router for each incoming packet. The enqueue procedure is shown in Algorithm 4.1 and its described as follow:

The enqueue procedure starts when NDN router r receives a packet under the name prefix p through interface f in coming rate $X_p(t)$, router r will first follow the NDN forwarding plane by checking if can satisfy the Interest packet router from the router cache; otherwise it will forward it to PIT to record its incoming interface. The router then checks the FIB to get outgoing interface f and send it to the enqueue procedure of that interface. If the router receives a Data packet, it checks the interfaces that request Data packets in PIT and sends it to the enqueue procedure of each interface.

Algorithm 4.2 Prefix Shaping Time scheme

```
ShapedTime(Interestqueue, PrefixShaping)
#Check the Number of Data Queues Sojourn Time of prefix i
PrefixShaping = 0;
for(i = 0; i ≦ n; i + 1)
 if(ST_i < target) then
    if(PrefixShaping = 0) then
         PrefixShaping = ST_i; //*First assignation for Prefixshaping*//
    else
         if(PrefixShaping < ST_i) then
             PrefixShaping = ST_i; //*Choosing the least sojourn time*//
         endif
 else
    SendNack()
endfor
```

When the enqueue procedure receives a packet, it will check its header and if it is a new name prefix SDWRR will add a new queue to the interface, denoted as q_{f_p}. Accordingly, it assigns the queue to the list of Interest or Data queues and assigns the default weight w_p, deficit counter $DC_{q_p}$to each queue. If it is not a new name prefix, the received packet is forwarded to the available queue q_{f_p} if it does not exceed the time threshold τ or the queue is not full to enqueue.

4.2.1.2 Dequeue procedure

The main work of SDWRR takes place in the dequeue procedure as the scheduling and shaping part occurs when dequeuing the packet from the interface router to uplink or downlink. The procedure of dequeuing is shown in Algorithms 4.2 and 4.3 its described as follow:

First, SDWRR assigns a weight w_p to each queue using Equation 4.1 to check if the queue header is an Interest packet; the weight is assigned by Algorithm 4.2 and Equation 4.2 as it takes the sojourn time of the Data queue of the same prefix ST_{p_D} in the downlink queue interface q_{f_p}; if the header is a Data packet the weight will be $w_p = 1$.

$$w_p = \begin{cases} \frac{1}{S_p} & Header = InterestPacket \\ 1 & Header = DataPacket \end{cases} \tag{4.1}$$

The calculation of the Interest queue weight w_p is done by calculating the Data queue sojourn time of the same prefix in the downlink. Since Interest packets of the same prefix p may be sent from many consumers u and received through different interfaces f in router r, the NDN router procedure will forward the first Interest to uplink only as one flow prefix and store nonce and the incoming interface of the other in PIT; when a Data packet is received the router will copy it and forward it back to each downlink interface from which the Interest packet was received. Thus, for each downlink interface SDWRR creates a Data queue. As the Interest queue weight depends on the Data queue sojourn time, the weight is calculated by taking the minimum sojourn time of the Data queue set, as in Equation 4.2.

$$S_p = \begin{cases} ST_{p_D} & q_{f_p} = 1 \,\&\, ST_{p_D} < \tau \\ \min(ST_{p_D}) & q_{f_p} > 1 \,\&\, ST_{p_D} < \tau \\ NACK & ST_{p_D} > \tau \end{cases} \tag{4.2}$$

As in Equation 4.2 each Data queue q_{f_p} receives a packet it will check whether it exceeds time threshold τ, and the SDWRR continues the process of enqueuing to q_{f_p}. Otherwise SDWRR sends a NACK to the interface of the congested queue or drops the packets.

Algorithm 4.3 Shaping Deficit Weighted Round robin (*Dequeue Function)*

```
While(TRUE) do
   for(i = 0; i ≦ n; i + 1)
      if(Head(Queue_i) == InterestPacet)then
         w_i = 1 / ShapedTime_i
      else
         w_i = 1
      SumWeight+ = w_i;
   end for
   if(ActiveList not Empty && SumWeight > 0) then
      i = the index at the Head of ActiveList;
      Quantum_i = w_i / SumWeight * C_(t);
      DC_i = Quantum_i + DC_i;
         While(DC_i > 0 && Queue_i not empty) do
            PacketSize = Size(Head(Queue_i));
            if(PacketSize < DC_i) then
               Extract enqueue timestamp from packet Headr;
               dequeuetime = Time(Now);
               sorojurntime = dequeuetime − enqueuetime;
               Dequeue Packet from (Queue_i);
               if(Head(Queue_i == InterestPacket))then
                  DC_i = DC_i − DataPacketsize;
               else
                  DC_i = DC_i − Packetsize;
            else
         break; //*'skip while Loop *//
         if(Empty(Queue_i))then
            DC_i = 0
         else
            InsertAtiveList(i);
   endIf
endWhile
```


After calculating the weight for each separate queue, SDWRR starts dequeuing from each queue in interface f_n by summing the weight of each queue in the interface q_{f_p} as in Equation 4.3. After calculating the weight and summing it the SDWRR scheduling function will use round-robin servicing with a quantum of service assigned to each queue; the only difference from traditional round-robin is that if a queue was unable to send a packet in the previous round because its packet size was too large, the remainder from the previous quantum is added to the quantum for the next round. Thus, deficits are kept track of; queues that were shortchanged in one round are compensated in the next round. Therefore, signing a quantum is the main function to control the dequeue in SDWRR. SDWRR calculates quantum k that denote as $k_{q_p}$and deficit counter $DC_{q_p}$of each prefix queue in interface as Equation 4.4 and Equation 4.5. After assigning the quantum to each queue in the interface, SDWRR starts dequeuing packets from each queue in the interface.

$$W_{f_q} = \sum_{n}^{1} w_{q_p} \tag{4.3}$$

$$k_{q_p}(t) = \frac{w_{q_p}}{w_{f_q}} * C(t) \tag{4.4}$$

$$DC_{q_p} = k_{q_p} + DC_{q_p} \tag{4.5}$$

When deficit counter $DC_{q_p}$of each queue is calculated by Equation 4.5 the procedure of dequeuing begins by checking the packet size at the head of the queue; if it is less than or equal to the deficit counter the packet is transmitted uplink. The deficit counter then checks the head of the queue; if it is a Data packet the deficit counter will be reduced by the number of bytes in the dequeued packet and if it is Interest the counter will be reduced by the number of bytes of the related Data packet, as Equation 4.6.

$$DC_{q_p} = \begin{cases} DC_{q_p} - PacketSize & Header = DataPacket \\ DC_{q_p} - DataPacketSize & Header = InterestPacket \end{cases} \tag{4.6}$$

The rest of deficit counter subtraction is compared with the following packet, and if deficit counter subtraction is more than the packet size it will transmit again; otherwise, the subtraction will keep the deficit counter for the queue for the next round, and start to queue from the next queue consecutively. Before dequeuing a packet, the packet sojourn time $ST_{q_p}$of the queue q_{f_p}will be calculated by detaching the stamped time on the head of the dequeued packet and subtracting it from the time of dequeuing, as in Equation 4.7 and check if it is Interest header will update the ST_{p_I} else update ST_{p_D}.

$$ST_{q_p}(t) = T_{out} - T_{in} \tag{4.7}$$

4.2.2 SDWRR Verification and Validation

The main reason for conducting verification is to ensure that the proposed SDWRR is implemented properly in the ndnSIM simulation environment, and is programmed correctly in C++. As explained in Chapter Three, SDWRR was verified and a snapshot of the implementation using the Eclipse platform is shown in Figure 4.2. From the figure, it can be confirmed that SDWRR does not appear to contain bugs or errors and is programmed correctly.

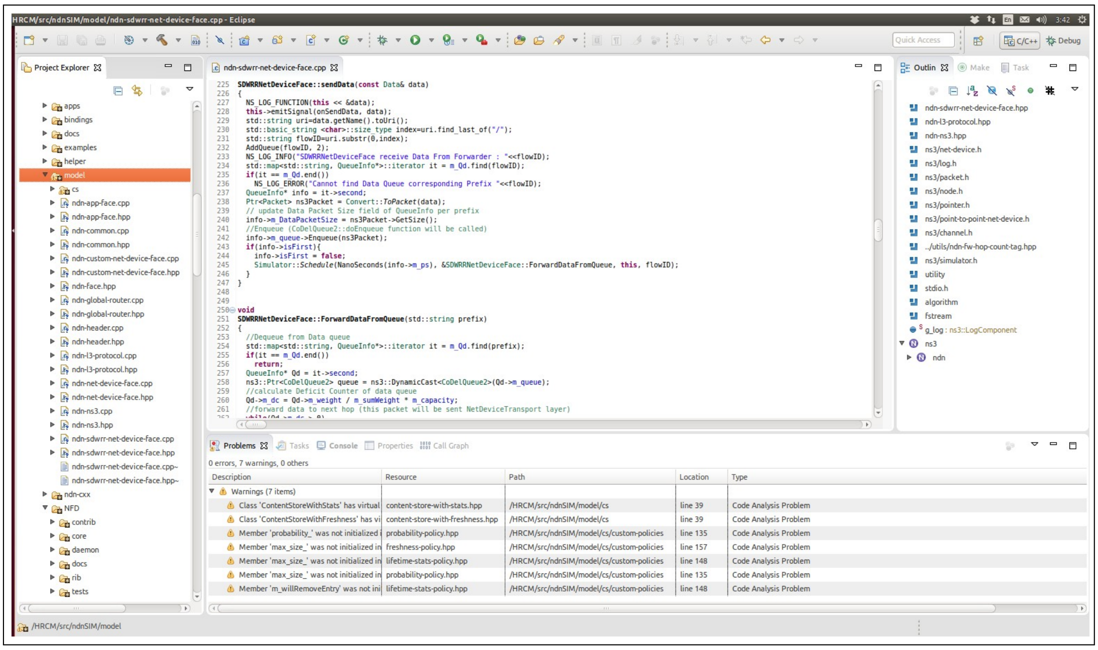


Figure 4.2. SDWRR Verification

To validate the accuracy of the proposed SDWRR scheme, a graphical comparison technique is used. In this technique, the graphs of results generated from simulating the model over time were compared with the graphs of results of valid scheme variables in order to investigate behaviours. The validation of SDWRR is to ensure that it meets the requirements, focusing on examining the rate adaptation and fairness in the simulation scenario and comparing them with the PCON scheme. Like SDWRR, PCON was designed using active queue management and feedback rate to control the forwarding rate.

Figure 4.3 illustrates the dumbbell topology used in the validation of SDWRR, conducted in ndnSIM [137]. Each consumer is associated with BIC conservative window adaptation to control the Interest rate on the consumer side and each router is associated with SDWRR/PCON scheme to control the forwarding rate. The router queue size is 1000 packets and simulation time 100s. For this simulation scenario, the results obtained from SDWRR were compared with results obtained from PCON. This com-

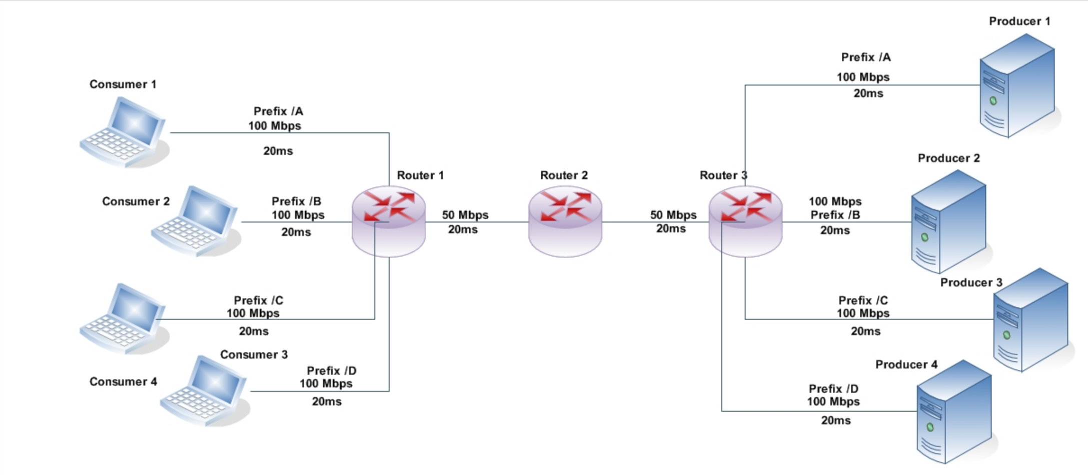
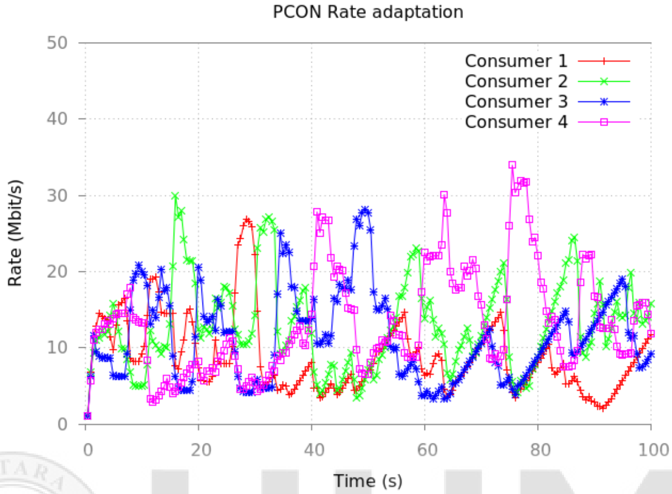


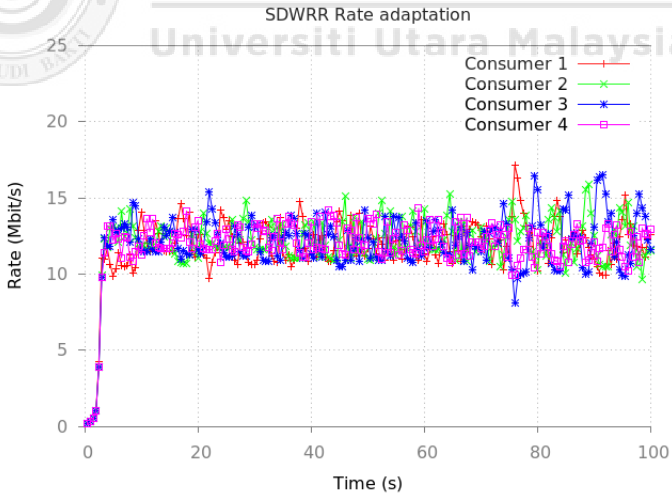
Figure 4.3. Dumbbell topology

parison was based on the rate adaptation behaviour and fairness of SDWRR. In other words, the SDWRR is expected to show different behaviour from PCON as SDWRR adds the scheduling behaviour.

In Figure 4.4 (b), the fluctuations in the SDWRR graph are not identical with those in the PCON graph Figure 4.4 (a). It can be observed that the rate of PCON has irregular fluctuations between consumers that reduce the fairness between them. In NDN the arrival time of data is not constant and PCON sends data on a First Come First Served (FCFS) basis, so consumers with low delay try to take more bandwidth than the others, affecting the sharing fairness as shown in Figure 4.5. Furthermore, as the control of congestion and rate in PCON is taken by consumers, this slows the reaction of rate adaptation between different consumers. On the other hand, in SDWRR, it can be observed that the rate fluctuations become significantly stable due to the scheduling ability of the SDWRR based on the delay of the interface queues conditions. Also, SDWRR's shaping control is done by observing the queue of each flow and assigning a fair quantum to each flow, as shown in Figure 4.5. Nevertheless, SDWRR is not affected by the variation in arrival times of the Data as the forwarding rate is shaped in a router along with the consumer. Hence, the Figures 4.4 and 4.5 shows that the overall behaviour of SDWRR corresponds with its description and analysis, supporting the



(a) PCON



(b) SDWRR

Figure 4.4. Validation Result (SDWRR v PCON Shaping)

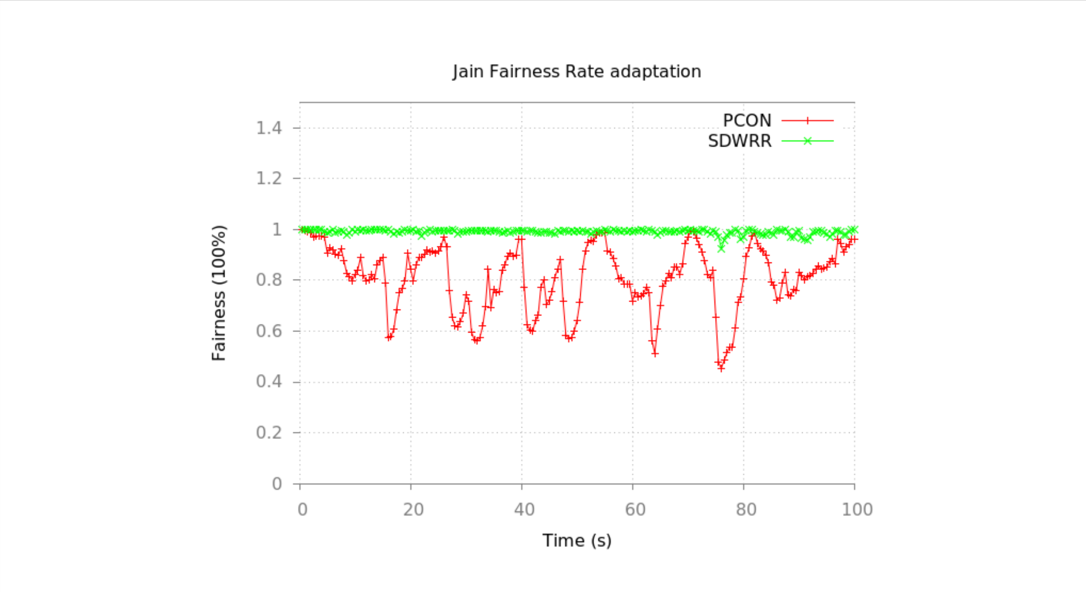


Figure 4.5. Validation Result (SDWRR v PCON Shaping Fairness)

validity of the SDWRR scheme.

4.3 Queue-delay Parallel Multipath forwarding strategy

The location-independent, Interest aggregation and in-network caching features proposed by NDN are a moving innovation in networking. Properties have rendered the abundant literature on congestion control, multi-path forwarding and fairness of the IP architecture as no longer compatible. Although there are many publications in the NDN context, they lack of NDN definitions. However, this forced many researchers to use the TCP/IP context. The unpredicted Interest packets aggregation within the networks and the significant variation in RTT measurements, because of the in-network caching, stop network to perform sufficient fairness, multi-path forwarding and managing the dynamics in return Data packets. For that, to create a sufficient multi-path forwarding strategy, have to overcome two obstacles. First, it is required to set the NDN flow independently from the source to the destination. Second, it is mandatory to control the end-user consumer's fairness and to utilize approaches without depending on RTT computations.

4.3.1 QPM Design

Queue-delay Parallel Multipath forwarding scheme (QPM) proposed to utilize all available bandwidth on the link and maintain fairness between end users without relying on RTT. This maximizes end-user throughput without affecting the fairness between different prefix flows. QPM uses the multiple queues built by SDWRR in each router interface (one for each active prefix name) and the packet sojourn time of these queues to manage the split ratio of the incoming flow rate of a certain prefix.

As Algorithm 4.4 shown, when router r receives Interest packet with prefix p through interface f in rate X_p is not satisfied by the local cache or by a pending request record in the PIT, the Interest packet will be handed to the proposed scheme to check available outgoing interfaces for the Interest name prefix in FIB; it sends this to SDWRR to create a queue for this named prefix in each available interface. QPM then calculates the split ratio s_{p_f} to each interface by dividing the incoming rate into each face equally, as in Equation 4.8.

$$s_{p_f} = \frac{x_{p_i}}{n} \tag{4.8}$$

After sending the Interest packet to the available interface the SDWRR scheme maps it to its prefix Interest queue and hands it to the queuing scheme to enqueue it. Before enqueuing the packet, the queuing scheme checks if there is space in the Interest queue and that the sojourn time of the packets in the queue does not exceed the threshold. The scheme stamps the time at the head of the packet and enqueues it; however, if the queue is full the packet will be dropped or if it is above the threshold the weight of the queue will be equal to zero. When SDWRR visits the prefix queue to forward the packets to the link the queuing scheme dequeues the packet and calculates its sojourn

Algorithm 4.4 Parallel Forwarding Strategy

$n = AvilableInterface;$
$p = Prefixname$
$f = Interface$
$p = ExtractFlow(I)$
$On\,arrival\,Interest\ packet\ p\,of\ flow\,I$
$if(ExistsInInterestList(p) == FALSE)then$
  $InsertInterestlist(p);$ //*add p to Interest list*//
  $for(f = 0; f \leqq n; f++)$
  $s_{p_f} = \frac{x_p}{n}$
$else$
#Check the InterestQueues Weight of prefix i
$for(f = 0; f \leqq n; f++)$
$if(ST_{q_p} < target)$
  $if(DataMarkFlag > 0)then$
    $count = DataMarkFlag;$
    $w_i = (\frac{1}{ST_{qp}} - (\frac{1}{ST_{qp}}(\alpha * count));$
  $else$
    $count = 0;$
  $if(Nack == TRUE)then$
    $count = count + 1;$
    $w_{f_p} = (\frac{1}{ST_{qp}}(\frac{1}{ST_{qp}}(\beta * count));$
  $else$
    $count = 0;$
    $w_{p_f} = \frac{1}{ST_{qp}};$
$else$
  $w_{f_p} = 0;$
$SumWeight += w_{f_p};$
#Rate split calculation of prefix i
  $if(SumWeight > 0)$
    $for(p = 0; p \leqq n; p + 1)$
      $if(w_{f_p} > 0)$
        $S_{p_f} = \frac{w_{fp}}{SumWeight} * X_p;$
      $else$
        $S_{p_f} = 0;$
  $else$
  $MarkPIT$
$end\ if$

time.

When the next Interest packets with the same prefix arrive at the router and it already has queues in the uplink face, the scheme calculates the split ratio to each interface by calculating the weight of the related queue in each available interface. The weight is the premier factor in the scheme as the split ratio and the congestion calculation depend on the packet sojourn time of each Interest queue in each interface for the same prefix. However, if a certain interface receives marked data or NACK for a specific prefix the forward weight of the queue is decreased each time, and increases the other queue until the weight of all the queues equals zero. The scheme then marks the data of that prefix and sends it to decrease the forwarding rate in downlink routers.

Thus, the weight indicates the sojourn time of each prefix queue if it is below the threshold for processing the calculation; if not, the weight will be set as zero as in Equation 4.9. The weight calculation function also checks the prefix Interest queue flag in the PIT if it has received marked data or NACK, to calculate the weight which decreases by α each time it receives mark data and β each time it receives NACK; otherwise the weight is calculated normally as in Equation 4.10. The weight scheme uses marked data and NACK received from uplink routers to decrease the split ratio to a certain prefix queue in order to control the prefix queue congestion in uplink routers.

$$w_{f_p} = \begin{cases} M & ST_{q_p} < \tau \\ 0 & ST_{q_p} > \tau \end{cases} \tag{4.9}$$

$$\mathrm{M} = \begin{cases} (\frac{1}{ST_{q_p}} - (\frac{1}{ST_{q_p}}(\alpha * n))) & \mu = Mark_D Received \\ (\frac{1}{ST_{q_p}} - (\frac{1}{ST_{q_p}}(\beta * n))) & \mu = NACK\,Received \\ \frac{1}{ST_{q_p}} & \mu = 0 \end{cases} \tag{4.10}$$

After calculating the weight $w_{f_p}$of each Interest queue for the same prefix p QPM sums these weights W_{q_p}as in Equation 4.11. Therefore, the Interest split rate s_{p_f} to each interface is calculated by multiplying the Interest rate $X_p$with the normalizing sum of the prefix queue weight, as in Equation 4.12.

$$W_{q_p} = \sum_{n}^{1} w_{f_p} \tag{4.11}$$

$$s_{p_f} = \begin{cases} \frac{w_{fp}}{W_{qp}} * X_{p_i} & W_{q_p} > 0 \\ MarkData & W_{q_p} = 0 \end{cases} \tag{4.12}$$

$$x_{p_i} = \sum_{f=1}^{n} s_{p_f} \tag{4.13}$$

The scheme indicates queue congestion by the weight of the prefix queues and controls it by changing the forward rate to other available faces; if the congestion is not controlled, the scheme will signal downlink to decrease the flow rate by marking the Data packet if the congestion is in the Interest queues and NACK if the congestion is

Algorithm 4.5 Data Pipeline

$Function\ OnData(dataPkt, incomingFace, fibEntry, pitEntry)$
$if \quad dataPkt.isMarked()\ then$
$\quad DataMarkFlag += 1$
$else$
$\quad DataMarkFlag = 0;$
// For each Downstream Face
$for\ dsFaceIN\ pitEntry \quad do$
$\quad if \quad pitMarked \quad then$
$\qquad MarkOutgoingData()$
$\quad else$
$\qquad ForwardData()$
$endfor$
$endfunction$

in the Data queues. Because of the different procedures for Data and Interest packets in NDN, QPM uses two different forms of signalling. When the Interest queue is congested all downlink interface which requested the same Interest packet will be affected and it needs to notify all of them by one signalling procedure; this is why we mark Data packets. However, when congestion is indicated in Data queues only the face of the queue will be affected by it, and we use NACK to notify downlink routers of that interface only.

For that, Algorithm 4.5 shows this modification to the Data packet pipeline and how the packet signalling is done. Algorithm 4.6 shows the QPM modification to the NACK pipeline procedure.

4.3.2 QPM Verification and Validation

The main reason for conducting verification is to ensure that the proposed QPM is implemented correctly in the ndnSIM simulation environment and is programmed correctly in C++. As explained in Chapter Three, QPM has been verified, and a snapshot of the implementation is shown in Figure 4.6 using the Eclipse platform. From the figure, it can be confirmed that QPM does not appear to contain bugs or errors and is

Algorithm 4.6 NACK Pipeline

Function $Process(NACK)$
$PitEntry = PIT.Find(NACK.Name)$
if $(PitEntry = 0$
$OR\ PitEntry.RetryTimerexpiredor$
$OR\ NACK.Nonce \nexists PitEntry.NonceList)$
$then$
$StopProcessing$
$NackFlag = 0;$
$else$
$NackFlag = NackFlag + 1;$
$end\,if$
$Forward(NACK.Interest, PitEntry)$
endfunction

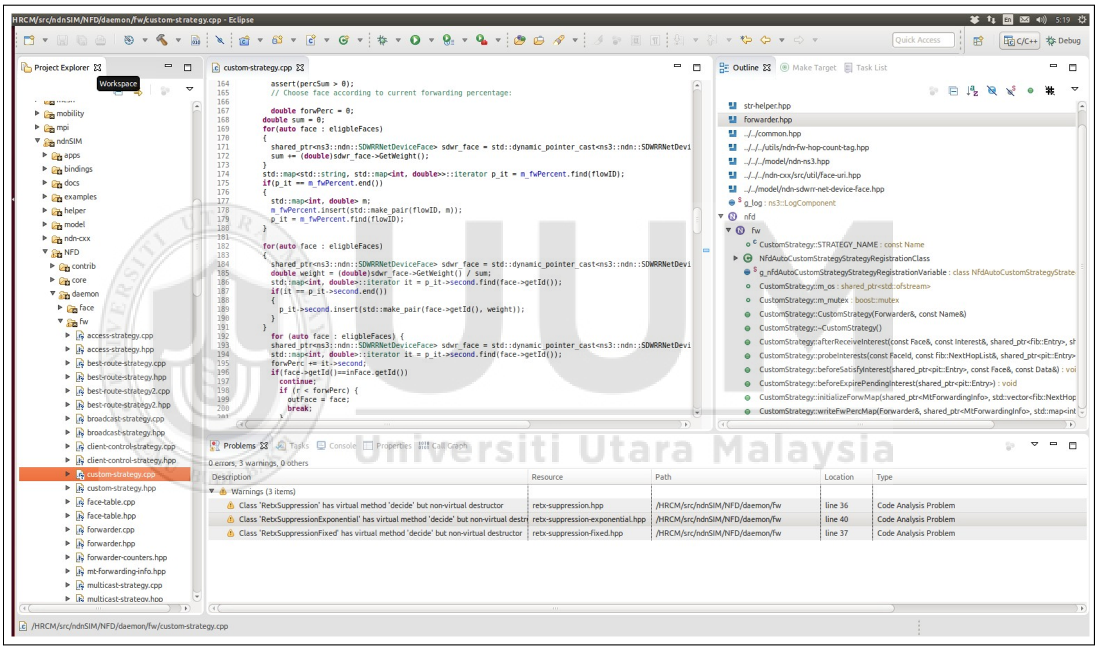


Figure 4.6. QPM Verification

programmed correctly.

To validate the accuracy of the proposed QPM scheme, the graphical comparison technique is used. The graphs of results generated from simulating the model over time are compared with the graphs of results of valid schemes. The validation of QPM is to ensure that it meets the requirements, focusing on examining the rate distribution in the simulation scenario. This rate distribution is compared with the PCON scheme

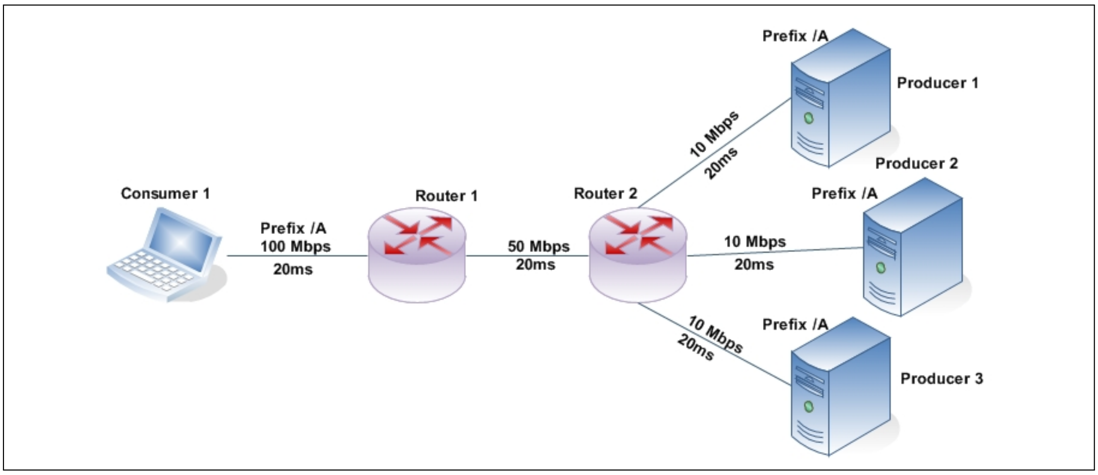


Figure 4.7. Multipath topology

because, like QPM, PCON was integrated with multipath functions.

Figure 4.7 illustrates the multipath topology used in the validation of QPM, and conducted in ndnSIM [137]. Each consumer is associated with BIC conservative window adaptation to control the Interest rate on the consumer side and each router is associated with QPM/PCON to control the rate distribution. Again, the router queue size is 1000 packets and simulation time 100s. For this simulation scenario, the results obtained from QPM were compared with those from PCON. This comparison is based on the rate distribution behaviour of QPM. In other words, the QPM scheme is expected to show different behaviour from PCON as QPM distributes the rate based on local parameters and at the beginning of the forwarding.

In Figure 4.8 (b), the fluctuations in the QPM graph are not identical to the PCON graph in Figure 4.8 (a). It can be observed that the distribution rate of PCON started forwarding the whole rate to producer 1, taking around 20ms to send to the three paths equally, even though it experiences some irregular fluctuations between the different paths. PCON forwards the arriving Interest packet first to the best path and then, if the first path did not satisfy the incoming Interest rate, it divides the rate to the second-best path and then to all available paths. PCON distributes the incoming rate based on the

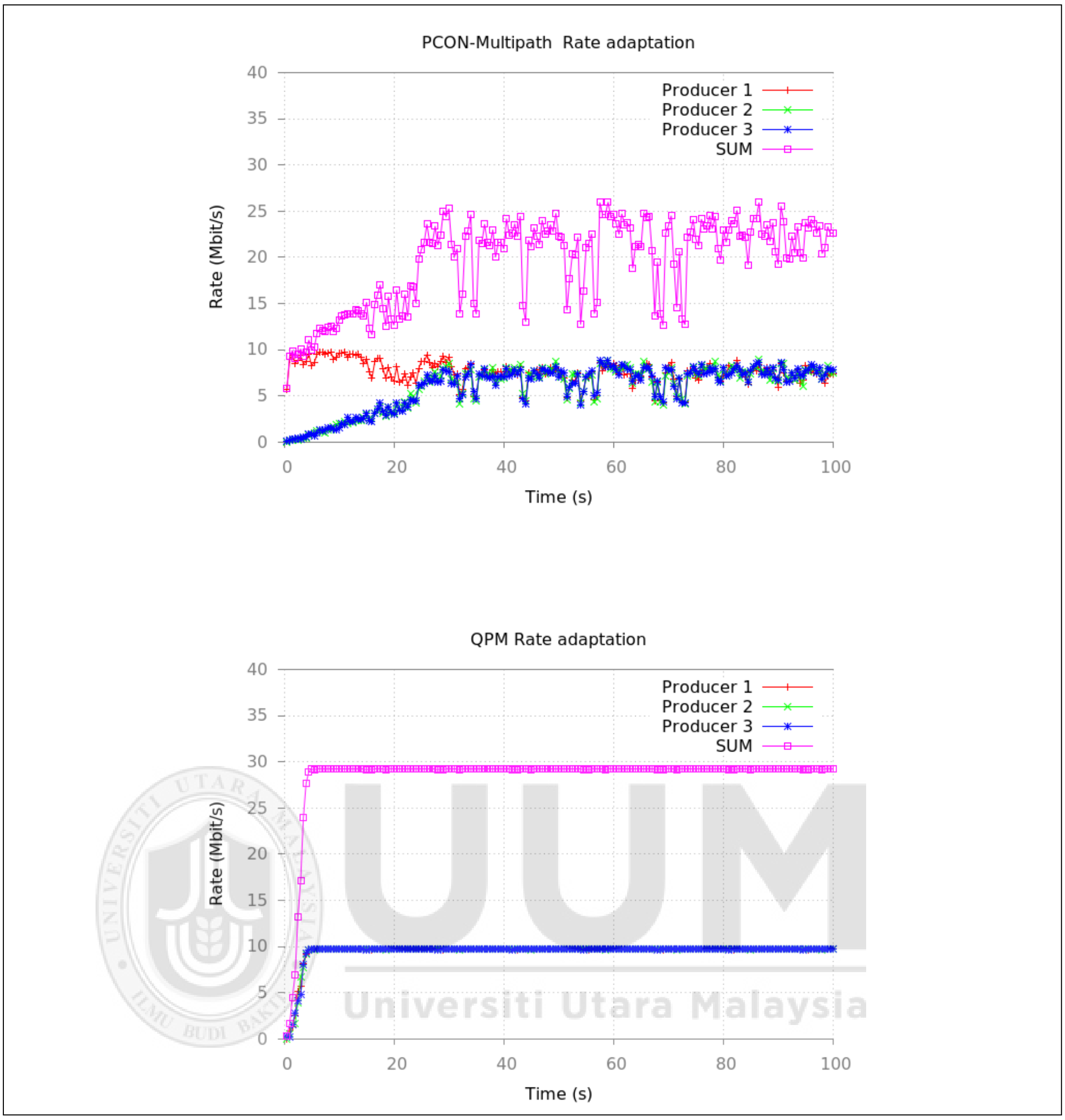


Figure 4.8. Validation Result (QPM v PCON Multipath Forwarding)

incoming marked data, delaying the distribution reaction and bandwidth utilization. As shown in Figure 4.8 (a), PCON only utilizes around 25Mbit/s of the 30Mbit/s link bandwidth. On the other hand, in QPM, it can be observed that the rate distribution is significantly stable and equal from the beginning of the simulation, because it distributes the Interest rate equally to all available paths. In addition, QPM depends on the local packets' sojourn time alongside the incoming marked Data packet or NACK to distribute the load between different paths to speed up the load distribution. Moreover, QPM utilizes nearly all the 30Mbit bandwidth available in the network, as shown in Figure 4.8 (b). Hence, the graphs show that the overall behaviour of QPM corresponds with its description and analysis, supporting its validity.

4.4 Explicit Control Agile-based Conservative Window Adaptation

When each consumer increases his/her rate above the available bandwidth in the network, too much data will be buffered in the queue of the bottleneck router, increasing congestion and data queue delay. Moreover, each consumer competes to increase their sending rate to satisfy their need, and congestion is unavoidable. Several consumer schemes adapt congestion avoidance mechanisms to control the Interest sending rate and to avoid congestion in uplink routers. Nevertheless, traditional end-to-end congestion control mechanisms do not fit the multi-source transports in NDN as both delay-based and loss-based controls are deployed merely on the senders which own the data objects. NDN should deploy congestion control on the consumer node, which requires data; in addition, a delay-based congestion control, such as Vegas [167] and BBR [168] adjusts their window size or send rate by estimating RTT value. This is not suitable to use in NDN because of multisource transport. Loss-based congestion control, such as Reno, BIC [169] and CUBIC [170] can react to duplicate ACK caused by packet loss because of single TCP transport packets and sequential ACKs. However, it is impossible in NDN to maintain the strict sequence of Data packets arriving from different producers or routers. Therefore, a congestion avoidance scheme is designed for NDN to exploit its distinctive transport properties.

As a consumer in NDN retrieves a Data packet by sending the Interest packet one to one, EC-Agile is designed to use the incoming Data packet to increase the forwarding rate and decrease it on receiving a congestion notification or time-out. EC-Agile adapts a scheme called Agile-SD that uses Agility Factor (AF) which reacts quickly to the changing rate between competitive consumers.

4.4.1 EC-Aglie Design

Communication in NDN starts by the consumer sending Interest packets to retrieve the needed content from uplink routers and producers. Algorithm 4.7 shows that when the EC-Agile scheme at consumer *u* receives a Data packet, it checks that the packet is not marked and increases the congestion window *cwnd* as Equation 4.14. On receiving marked data or NACK, EC-Agile checks if the scheme is not in the congestion avoidance stage and performs a major decrease using Equation 4.15 to give space for the congestion avoidance stage to control the increase. Otherwise, if EC-Agile receives marked data or NACK in the congestion avoidance stage the decrease will be lower to grab most of the available bandwidth, using Equation 4.16. Nevertheless, when time-outs occur EC-Agile performs at most one window decrease per packet timeout to prevent drastic decrease, as in Equation 4.17.

$$cwnd_i = cwnd_{i-1} + 1 \tag{4.14}$$

$$cwnd_i = cwnd_{i-1} * \beta_1 \tag{4.15}$$

$$cwnd_i = cwnd_{i-1} * \beta_2 \tag{4.16}$$

$$cwnd_i = cwnd_{i-1} - 1 \tag{4.17}$$

At the congestion avoidance stage, EC-Agile increases its congestion window *cwnd* by a small fraction after every reception of unmarked Data, as Equation 4.18. This fraction is calculated by Equation 4.19 where the fraction λ is the main parameter adapted from Agile-SD to calculate the increase in EC-Agile to utilize the link more accurately. To increase the bandwidth utilization, this fraction speeds up the growth of *cwnd* when the gap between it and the threshold is large and slows it down when the gap is small.

$$cwnd_i = cwnd_{i-1} + \frac{\lambda}{cwnd_{i-1}} \tag{4.18}$$

$$\lambda = \max(\frac{\lambda_{max} * gap_{current}}{gap_{totel}}, \lambda_{min}) \tag{4.19}$$

The fraction is calculated based on the gaps between *cwnd* and the last window from which the consumer received a notification, calculated in Equations 4.20 and 4.21. In addition, to ensure that the performance of EC-Agile is not less than the standard scheme, λ_{min} must be always set to 1 while λ_{max}must be always set to a value $\lambda \geq 1$. However, if λ_{max} is set to 1, EC-Aglie will behave exactly similar to slow star scheme. But, if it was set to a value > 1, such as 2, 3 or 4, the growth of *cwnd* will be fast in result improve the overall performance.

$$gap_{current} = \max((cwnd_{loss} - cwnd), 1) \tag{4.20}$$

$$gap_{totel} = \max((cwnd_{last} - cwnd_{Degraded}), 1) \tag{4.21}$$

Algorithm 4.7 Explicit Control Agile-based Conservative Window Adaptation

$Initialization:$
$\lambda_{min} = 1, \quad \lambda_{max} = 3$
$\beta_1 = 0.90, \quad \beta_2 = 0.95$
$cwnd = 1$
$Function\ OnData(dataPkt, seq)$
  $if\ HighData \leq seq \quad then$
    $HighData = seq;$
    $gap_{current} = \max((cwnd_{last} - cwnd), 1);$
    $gap_{totel} = \max((cwnd_{last} - cwnd_{Degraded}), 1);$
    $\lambda = \max(\frac{\lambda_{max} * gap_{current}}{gap_{totel}}, \lambda_{min});$
    $\alpha = \frac{\lambda}{cwnd};$
    $cwnd = cwnd + \alpha;$
  $endif$
  $if\ dataPkt.isMarked() \quad then$
    $if(cwnd < cwnd_{last}) \quad then$
        $cwnd_{last} = cwnd;$
        $cwnd = cwnd * \beta_1;$
    $else$
        $cwnd_{last} = cwnd;$
        $cwnd = cwnd * \beta_2;$
    $cwnd_{Degraded} = cwnd;$
    $ssthresh = cwnd - 1;$
  $endif$
endfunction
$Function\ OnTimeout()$
    $cwnd = cwnd - 1;$
    $ssthresh = cwnd;$
  $reset();$
endfunction

4.4.2 EC-Agile Verification and Validation

The main reason for conducting verification is to ensure that the proposed EC-Agile is adequately implemented in the ndnSIM simulation environment and is programmed correctly in C++. As explained in Chapter Three, EC-Agile has been verified, and a snapshot of the implementation is shown in Figure 4.9 using the Eclipse platform. From the figure, it can be confirmed that EC-Agile does not appear to contain bugs or

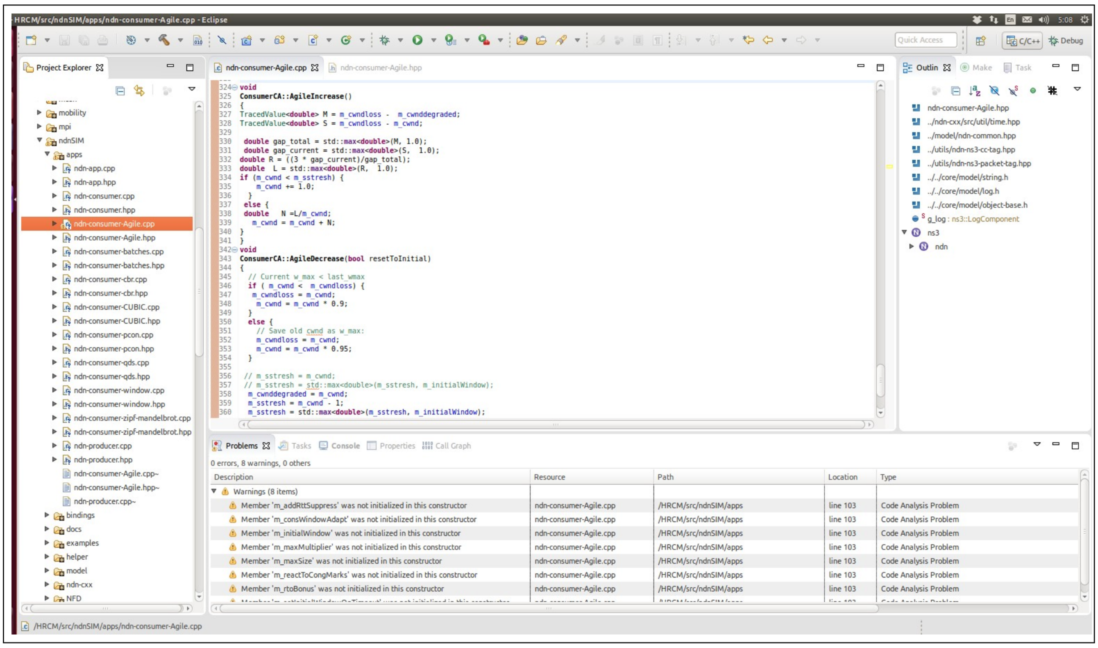


Figure 4.9. EC-Agile Verification

errors and is programmed correctly.

To validate the accuracy of EC-Agile, graphical comparison is used. The graphs of results generated from simulating the model over time were compared with the graphs of results of valid schemes. The validation of EC-Agile is to ensure that it meets the requirements, focusing on examining the rate adaptation and window size in the simulation scenario. These are compared with the BIC conservative window adaptation scheme because both schemes exhibit loss-based behaviour

Figure 4.3 illustrated the dumbbell topology used in the validation of EC-Agile, and conducted in ndnSIM [137]. Each consumer is associated with EC-Agile and BIC to control the Interest rate on the consumer side and each router is associated with BIC to send the feedback to consumers. Again, the router queue size is 1000 packets and simulation time is 100s. For this simulation scenario, the results obtained from EC-Agile were compared with those from BIC. This comparison was based on the Interest packet sending rate, and its adaptation behaviour. In other words, EC-Agile is expected

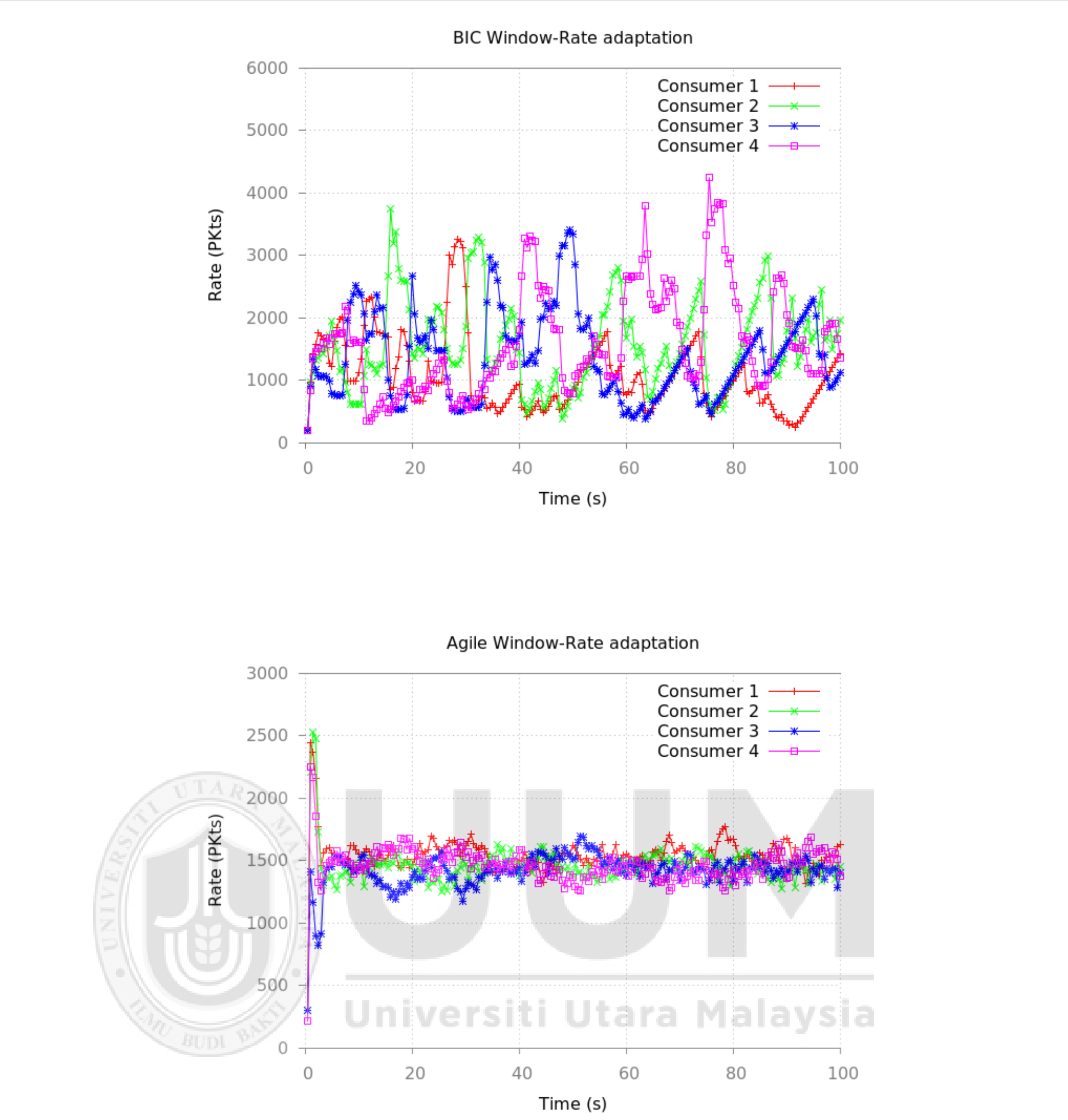


Figure 4.10. Validation Result (EC-Agile v BIC Conservative Window Adaptation)

to show different behaviour from BIC; its decreasing and increasing behaviour is not as aggressive as BIC's.

In Figure 4.10, the fluctuations in the EC-Agile's graph are not identical to those in the BIC graph. It can be observed that the rate of BIC has strict fluctuations because of the strict decrease when the consumer receives marked Data or NACK, while EC-Agile divides the window into two parts. In the first part it changes from the fast increase that occurs when the flow starts sending. The second part is the congestion avoidance, which occurs when the consumer receives the first congestion notification. When BIC receives a congestion notification, the window size drops to half and increases quickly, showing wide window fluctuation. However, in EC-Agile the drop is based on the β proposed in the scheme and increases by the fraction λ step by step until it reaches the full bandwidth.

4.5 Summary

This chapter introduced the SDWRR, QPM, and EC-Agile models for transport in a NDN environment. SDWRR, as a first contribution, was designed to shape forward flow, indicate congestion and ensure fairness by combining scheduling and queuing functions. The second contribution is the QPM scheme, designed to forward and adjust Interest packets in multiple paths to utilize all available bandwidth and maximize end-user throughput. EC-Agile, the third contribution, was designed to increase the forwarding rate in incoming Data packets and reduce it on receiving a congestion notification or time-out. These schemes are the main components of HRCM for controlling, monitoring and scheduling the transmission flow in NDN. They were described in this chapter together with the analytical model, pseudo code, verification and validation for each. The next chapter describes and evaluates the combination of the schemes to form HRCM.

CHAPTER FIVE
PERFORMANCE EVALUATION AND DISCUSSION

The design, implementation and validation of Hybrid Rate Control Mechanism (HRCM) schemes, as presented in Chapter Four, yielded positive results. In this chapter, the shemes have been integrated to as HRCM. The performance evaluation is presented to investigate the benefits of the features introduced in HRCM through extensive simulations with varying parameters and different topologies. For a better understanding and confirmation of HRCM, its performance is compared with PCON and HIS. The chapter starts with an overview and implementation structure of HRCM in Section 5.1. Results of the performance evaluations of HRCM compared with PCON and HIS are discussed in Section 5.2. Finally, a summary of the chapter is presented in Section 5.3.

5.1 Hybrid Rate Control Mechanism

The aim of this research was to develop and implement HRCM for NDN transport control. HRCM monitors, shapes and controls each incoming flow as well as indicating congestion and sending notifications for managing consumers' rates of flow, leading to better link utilization, stability and fairness. HRCM incorporates three components, SDWRR, QPM and EC-Agile, as presented in Chapter Four, to mitigate congestion and enhance forwarding. The research problem, goals specification, and research focus guided the design of the conceptual framework of HRCM.

SDWRR, QPM and EC-Agile were integrated into HRCM as a single mechanism in the NDN consumer and router code see Figure 5.1 and 5.2, and implemented in the ndnSIM environment. Incorporating HRCM components provides a fundamentally new direction for transport control in the NDN router, significantly mitigating congestion and enhancing forwarding performance as well as improving the entire network performance. Each step in HRCM addresses a particular issue in transport control in NDN, as follows.

In the first stage, when communication is started by the consumer, EC-Agile starts to send Interest packets to uplink routers. As soon as the NDN router receives the first Interest packet from the consumer, it records it in PIT and hands it to SDWRR. SDWRR checks for available interfaces in FIB that can satisfy the Interest packets and builds Interest and Data queues in each available interface. SDWRR then time stamps the Interest packets and enqueues them; during dequeuing, it takes the stamped time and calculates the sojourn time in each interface's Interest queue. When Data packets are received from the producer, SDWRR checks the number of interfaces requesting the Data packet in PIT. It then adds them to the Data queue of each interface and stamps the enqueue time. When dequeuing Data packets, SDWRR takes the stamped time and calculates the sojourn time for the Data queues.

In the second stage, when the consumer receives a Data packet, EC-Agile increases the Interest packet rate if there is no NACK or the Data packet received is not marked. As soon as the Interest packet is received by the router, QPM takes the sojourn time of each queue in each interface and calculates the forwarding percentage of each interface. SDWRR then adds Interest packets to each Interest queue based on the QPM percentage. However, if the sojourn time of any queue rises above the threshold, SDWRR checks if the indication on the Data queue NACK will be generated downlink to decrease its flow rate.

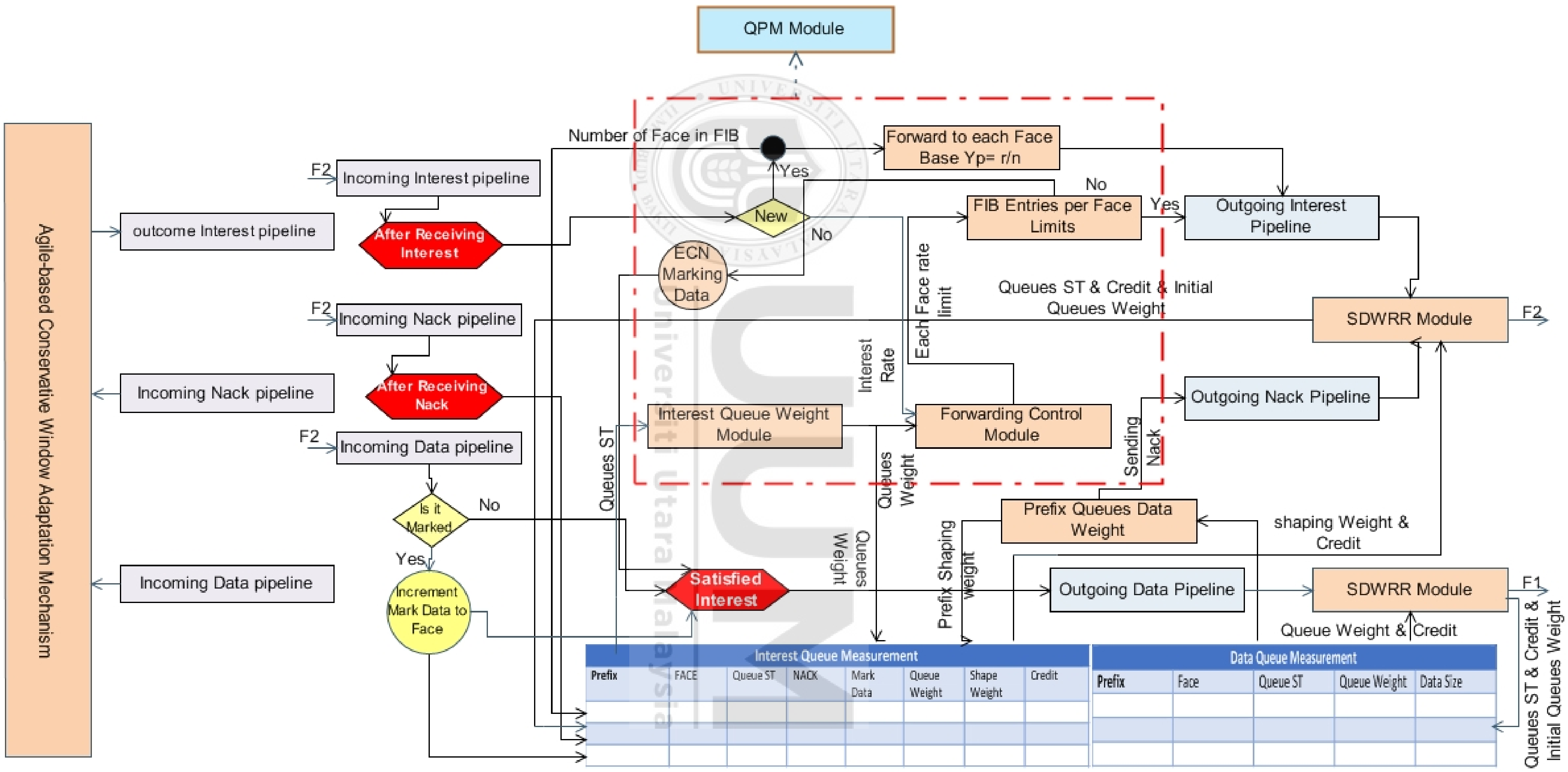


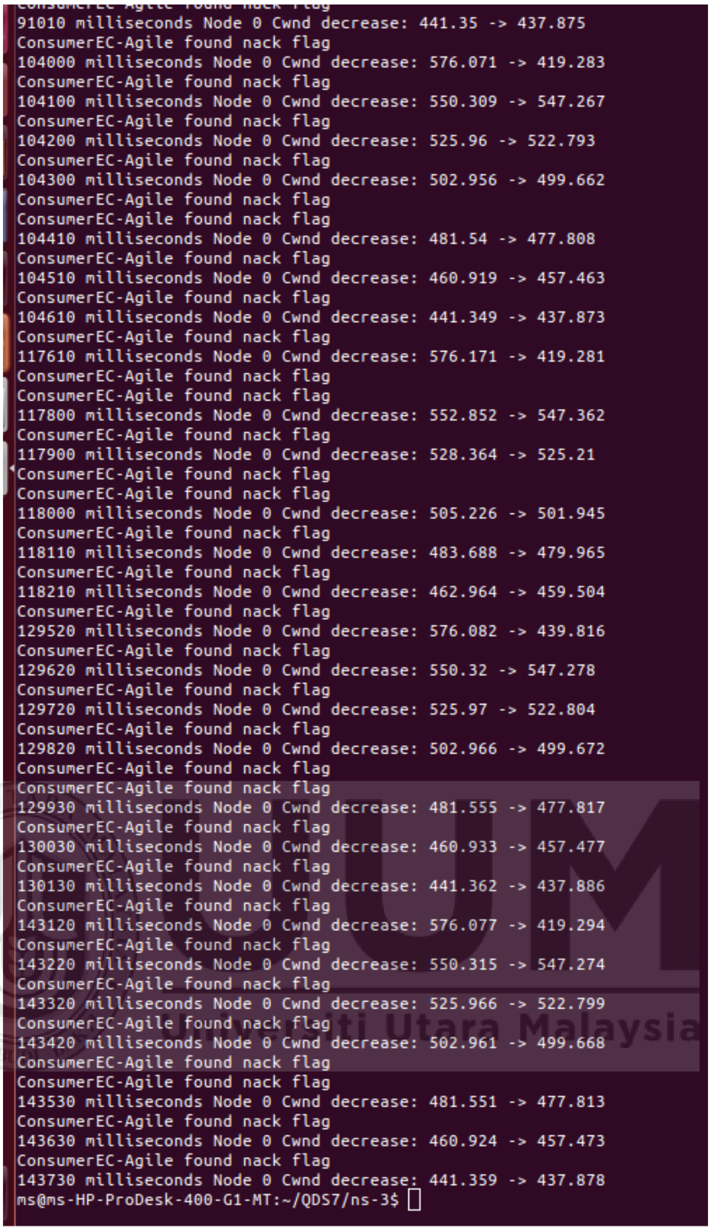
Figure 5.1. HRCM Conceptual Model

```
namespace ns3 {


NS_LOG_COMPONENT_DEFINE ("test-propo-mach-scenarioline");
//Callback function called when NdnStackHelper create router's face
shared_ptr<ndn::NetDeviceFace>
callBackCreateCustomFace(Ptr<Node> node, Ptr<ndn::L3Protocol> ndn, Ptr<NetDevice> netDevice)
{
  NS_LOG_INFO("Creating CustomFace for Router on node :"<<node->GetId());
  auto face= make_shared<ndn::SDWRRNetDeviceFace>(node, netDevice);
  face->setMetric(1);
  face->SetTarget("2ms");
  ndn->addFace(face);
  NS_LOG_INFO("Router " << node->GetId() << ": added Face as face #"
                       << face->getLocalUri()<<" face id ="<<face->getId());
  return face;
}
//Install custom Ndn to Routers

void
InstallNDNToRouter(NodeContainer& c)
{
  NS_LOG_INFO("Installing Custom NDN stack To Routeres");
  for(NodeContainer::Iterator it = c.Begin(); it != c.End(); it++)
  {
    SimHelper::setNodeQueue(*it, 100, "QDS", "5ms");
  }
  ndn::StackHelper ndnHelper;
  ndnHelper.SetDefaultRoutes(false);
  ndnHelper.SetOldContentStore("ns3::ndn::cs::Lru", "MaxSize", "200000");
  ndnHelper.UpdateNetDeviceFaceCreateCallback(ns3::PointToPointNetDevice::GetTypeId (), MakeCallback (&callBackCreateCustomFace));
  ndnHelper.Install(c);
  NS_LOG_INFO("Choosing QPM Stratgegy For Routers");
  ndn::StrategyChoiceHelper::Install(c, "/", "/localhost/nfd/strategy/QPM-strategy/%FD%06");

int
main(int argc, char* argv[])
{
 //consumer
    ndn::AppHelper consumerHelper("ns3::ndn::ConsumerECAgile");
    consumerHelper.SetAttribute("MinRto", UintegerValue(MIN_RTO));
    consumerHelper.SetAttribute("Size", StringValue("1000M"));
    for (uint32_t i = 0; i < consumers.size(); i++) {
      consumerHelper.SetPrefix("/dist"+std::to_string(i+1));
      consumerHelper.Install(consumers[i]);
    }
  int MIN_RTO = SimHelper::getEnvVariable("MIN_RTO", 300);

        LogComponentEnable("ndn.SDWRRNetDeviceFace", LOG_LEVEL_INFO);

        /*================================= Setup Network Topology ===================================*/
        AnnotatedTopologyReader topologyReader("", 25);
        topologyReader.SetFileName("src/ndnSIM/examples/topologies/topo-test-propo-mach-scenarioline.txt");
        topologyReader.Read();

        // Get Nodes uing topology reader
        NodeContainer consumers;
```

Figure 5.2. HRCM-test scenario

If the indication is in the Interest queue, QPM first checks if there is more than one Interest queue for the flow to decrease the forwarding percentage in the congested queue and increase it in the others. If all of them are congested, QPM marks each related Data packet received. When downlink routers in HRCM receive NACK or marked Data, QPM reduces the forwarding rate of the congested uplink by forwarding the rate to other interfaces. If all routers downlink are unable to handle the congestion, EC-Agile on the consumer side receives the NACK or marked Date packet and decreases its Interest rate.

To carefully evaluate HRCM, the proposed schemes were implemented in the official NDN simulator (i.e., ndnSIM). The simulations were carefully set to match the specifications of this study. The evaluation was therefore conducted to study the impact of HRCM on performance based on SDWRR, QPM and EC-Agile, as well as to



```
91010 milliseconds Node 0 Cwnd decrease: 441.35 -> 437.875
ConsumerEC-Agile found nack flag
104000 milliseconds Node 0 Cwnd decrease: 576.071 -> 419.283
ConsumerEC-Agile found nack flag
104100 milliseconds Node 0 Cwnd decrease: 550.309 -> 547.267
ConsumerEC-Agile found nack flag
104200 milliseconds Node 0 Cwnd decrease: 525.96 -> 522.793
ConsumerEC-Agile found nack flag
104300 milliseconds Node 0 Cwnd decrease: 502.956 -> 499.662
ConsumerEC-Agile found nack flag
ConsumerEC-Agile found nack flag
104410 milliseconds Node 0 Cwnd decrease: 481.54 -> 477.808
ConsumerEC-Agile found nack flag
104510 milliseconds Node 0 Cwnd decrease: 460.919 -> 457.463
ConsumerEC-Agile found nack flag
104610 milliseconds Node 0 Cwnd decrease: 441.349 -> 437.873
ConsumerEC-Agile found nack flag
117610 milliseconds Node 0 Cwnd decrease: 576.171 -> 419.281
ConsumerEC-Agile found nack flag
ConsumerEC-Agile found nack flag
117800 milliseconds Node 0 Cwnd decrease: 552.852 -> 547.362
ConsumerEC-Agile found nack flag
117900 milliseconds Node 0 Cwnd decrease: 528.364 -> 525.21
ConsumerEC-Agile found nack flag
ConsumerEC-Agile found nack flag
118000 milliseconds Node 0 Cwnd decrease: 505.226 -> 501.945
ConsumerEC-Agile found nack flag
118110 milliseconds Node 0 Cwnd decrease: 483.688 -> 479.965
ConsumerEC-Agile found nack flag
118210 milliseconds Node 0 Cwnd decrease: 462.964 -> 459.504
ConsumerEC-Agile found nack flag
129520 milliseconds Node 0 Cwnd decrease: 576.082 -> 439.816
ConsumerEC-Agile found nack flag
129620 milliseconds Node 0 Cwnd decrease: 550.32 -> 547.278
ConsumerEC-Agile found nack flag
129720 milliseconds Node 0 Cwnd decrease: 525.97 -> 522.804
ConsumerEC-Agile found nack flag
129820 milliseconds Node 0 Cwnd decrease: 502.966 -> 499.672
ConsumerEC-Agile found nack flag
ConsumerEC-Agile found nack flag
129930 milliseconds Node 0 Cwnd decrease: 481.555 -> 477.817
ConsumerEC-Agile found nack flag
130030 milliseconds Node 0 Cwnd decrease: 460.933 -> 457.477
ConsumerEC-Agile found nack flag
130130 milliseconds Node 0 Cwnd decrease: 441.362 -> 437.886
ConsumerEC-Agile found nack flag
143120 milliseconds Node 0 Cwnd decrease: 576.077 -> 419.294
ConsumerEC-Agile found nack flag
143220 milliseconds Node 0 Cwnd decrease: 550.315 -> 547.274
ConsumerEC-Agile found nack flag
143320 milliseconds Node 0 Cwnd decrease: 525.966 -> 522.799
ConsumerEC-Agile found nack flag
143420 milliseconds Node 0 Cwnd decrease: 502.961 -> 499.668
ConsumerEC-Agile found nack flag
ConsumerEC-Agile found nack flag
143530 milliseconds Node 0 Cwnd decrease: 481.551 -> 477.813
ConsumerEC-Agile found nack flag
143630 milliseconds Node 0 Cwnd decrease: 460.924 -> 457.473
ConsumerEC-Agile found nack flag
143730 milliseconds Node 0 Cwnd decrease: 441.359 -> 437.878
ms@ms-HP-ProDesk-400-G1-MT:~/QDS7/ns-3$
```


Figure 5.3. HRCM- results snapshot

compare performance with the PCON and HIS mechanisms. The output results of the HRCM-test scenario are illustrated in Figures 5.3.

5.2 Performance Evaluation of HRCM

Evaluation is important in research, particularly with comparative analysis of several designs to determine which design outperforms the others. This study uses the simulation method since it is mainly used for evaluation in many of the reviewed studies.

The main goal of these evaluations is to test the ability of HRCM to control forwarding and avoid congestion in NDN, as compared to PCON and HIS. The aim was not to measure HRCM's performance on a particular workload captured from a real network. Rather, it was to measure its performance under a range of network conditions and scenarios. For fair evaluation of the transport issue, there is no general agreement on the particular topology selected [139]. The network topologies chosen here were used by the researchers listed in Chapter 3: Baseline (small topology), Dumbbell (bottleneck topology) and Abilene (large-scale topology) with a variety of Data packet sizes, different start times and multiple content producers. The key step in all performance evaluation is selection of the performance metrics, although researchers consider that they have different meanings according to the situation in which they are used. In addition, different metrics used in different scenarios give us a complete view of the performance of the proposed mechanism. This study focuses on throughput, packet delay, link utilization, download time, queue length and Jain fairness metrics to measure performance against the objectives of this research.

5.2.1 Baseline Topology

Baseline topology is a common topology used in many network simulations. Several researches [38, 143, 144, 145] have used it for evaluating the window size, queue overflow and download time in CCN/NDN. It has been used to study the impact of competing flow rates, link utilization and fairness between consumers and publishers. A simple Baseline topology see Figure 3.8 and Table 5.1.

Table 5.1
Baseline Simulation Parameters

Parameter	Description
Number of Consumer	1
Number of producer	1
Number of Router	3
Number of links	4
Link Delay	10ms
Consumer Link Bandwidth to Router	100Mbit/s
Producer to Router	100Mbit/s
Between Routers	10Mbit/s
Simulation Time	100s

The throughput of the baseline topology is measured by observation of the three mechanisms, proposed HRCM and benchmark PCON and HIS. Figure 5.4 shows the performance as represented by packet rate against simulation time. The performance improvement of HRCM over PCON is approximately 15%, and for HIS nearly 2% of packet delivery. The difference between HRCM and HIS is negligible because both machines have congestion flow control inside the router. PCON's comparatively poor is due to the accumulated delay when congestion is detected; PCON sends the Interest packet back to the consumer, increasing the delay. As shown in the Figure 5.4, the throughput of PCON is not stable because the window size drastically reduces the flow when a consumer receives notification of congestion in the router. It then suddenly increases the sending flow until another congestion notification is received. That is, the performance of HRCM is better than PCON and nearly same as HIS with the baseline topology.

Figure 5.5 represents the time taken for a consumer to download file sizes of 100, 200 and 300 MB respectively. The time for the download is monitored for both proposed and benchmark solutions and presented in the graph of time in seconds against the file size in megabytes. The overall result shows that PCON has the poorest performance,

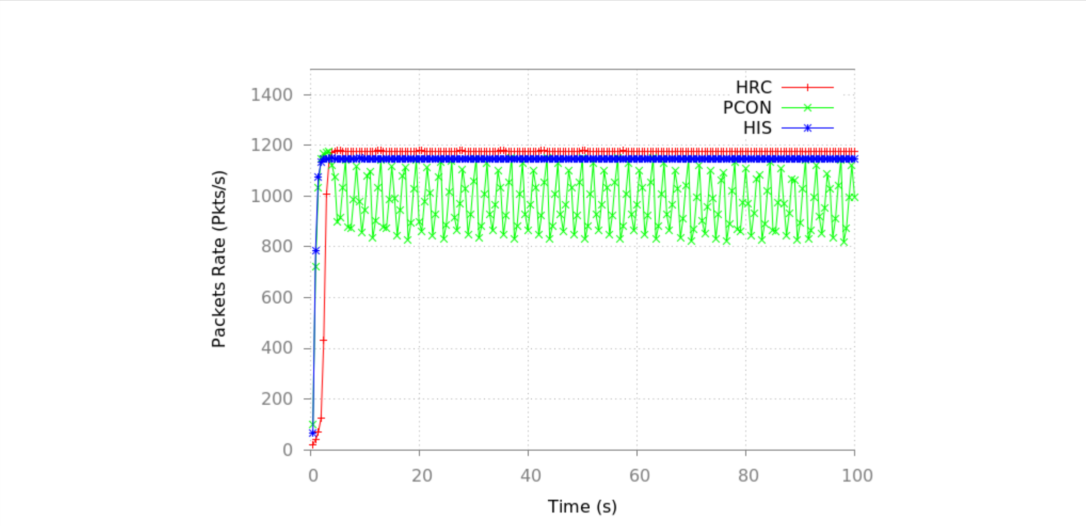


Figure 5.4. Baseline Throughput

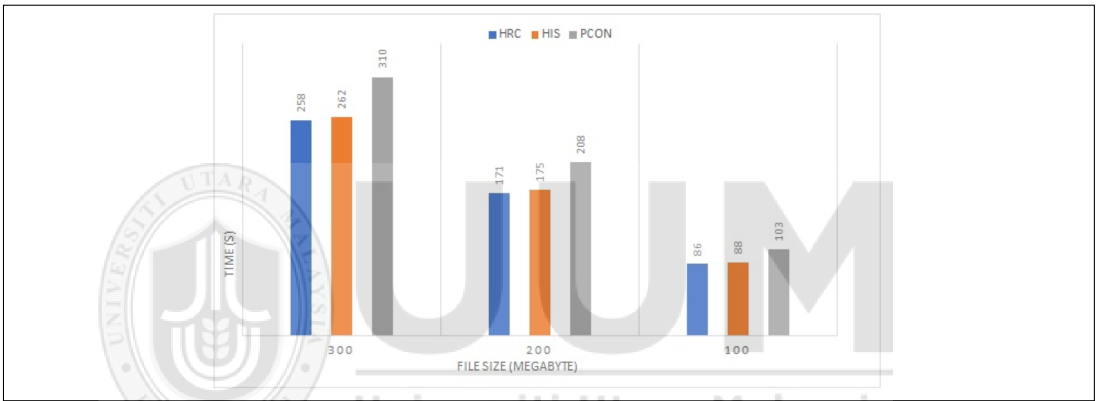


Figure 5.5. Baseline Download Time

which is not surprising given the throughput shown in Figure 5.5. PCON takes longer time to download the three files; the time for HIS is closer to that HRCM. HRCM performed 20% better than PCON and 6% better than HIS.

Figure 5.7 is a graph of packet delay in seconds against three simulation times: 0 to 10 sec, 10 to 25 sec and 25 to 100 sec. In the first situation observed, 0 to 10 sec of simulation time, the performance of HRCM is 20% higher than that of HIS and only 2% higher than PCON as the packet delays are unstable. With 10 to 25 sec of simulation time, the performance of HRCM is stable at 0.08 sec of packet delay, as is HIS, whereas PCON keeps on fluctuating between 0.08 and 0.09 sec. In the third situation,

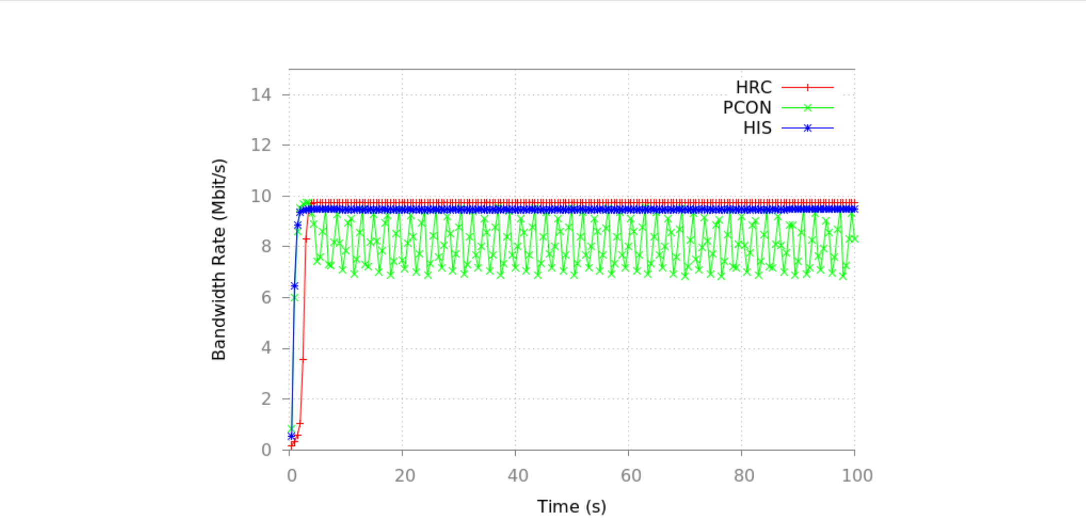


Figure 5.6. Baseline Link Utilization

25 to 100 sec of simulation time, HRCM maintained the average delay of 0.08, while PCON and HIS were unstable between 0.08 to 0.09 and 0.09 to 0.12 respectively. The result of this third situation shows the moderate improvement of HRCM, performing 12% better than PCON and 28% against HIS. Overall, average performance for the complete simulation time shows that HRCM has minimal packet delay compared with PCON and HIS . The worst performance by HIS is due to its queue length policy to determine congestion; while the queue is elongated the packet delay increases. Both HRCM and PCON use queue packet delay policies to determine the congestion: as long as the queue increases the notification of congestion is sent to the consumer without waiting until the queue is full, as is the case with HIS. Conclusively, HRCM is better than PCON, especially at 10 to 100 sec of simulation time, because HRCM utilizes two notifications of packet delay that actively predict congestion for both Data and Interest packets.

The queue length for the baseline topology is represented in Figure 5.8, with the three mechanisms plotted by number of packets on the y-axis against the simulation time on the x-axis. The queue-length performance of PCON appears stable from the beginning of the simulation until the end, within the packet range of 0 to 10 across the simulation

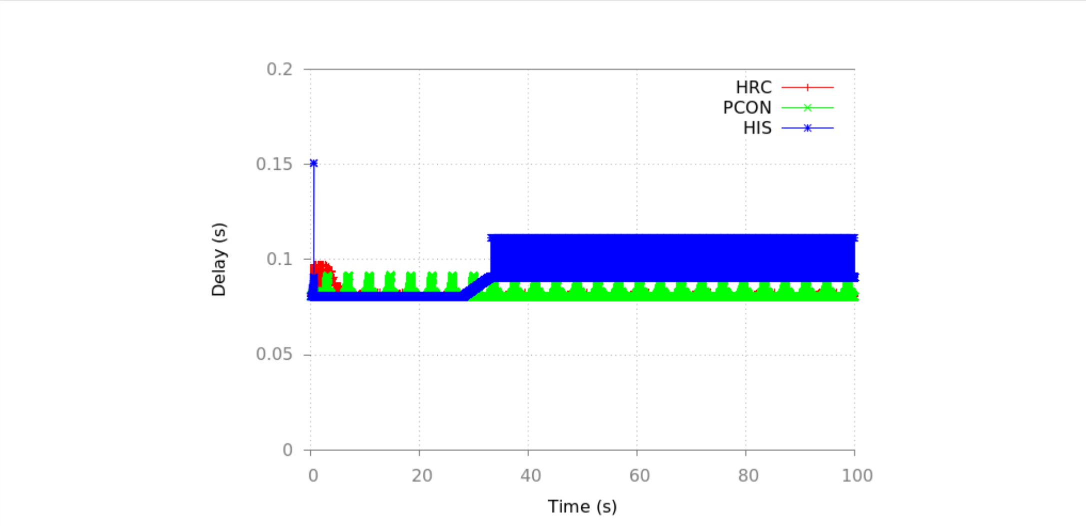


Figure 5.7. Baseline Delay

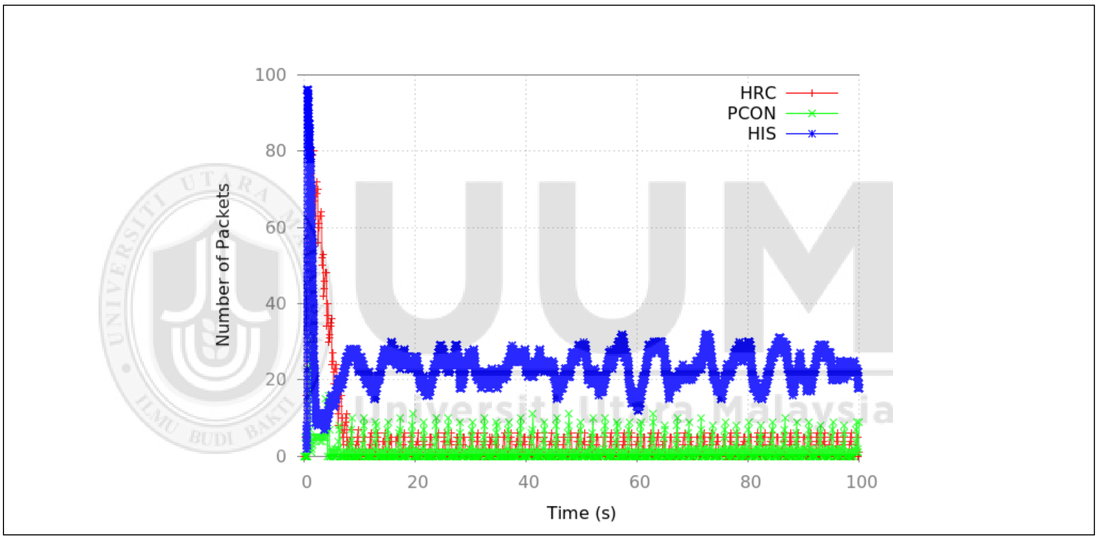


Figure 5.8. Baseline Queue Length

time. HIS approached 100 packets and then gradually reduced in stability to 15 to 30 packets at around 10 sec of simulation time. HRCM performed between the characteristics of PCON and HIS, with the number of packets nearly reaching the point attained by HIS and then descending to the lower level of PCON. Beyond 10 sec of simulation time, HRCM shows more desirable performance, improving by 55% against PCON and 77% against HIS. This result suggests that the dual behaviour of HRCM imitates the function of PCON packet delay and the function of the packet drop of HIS.

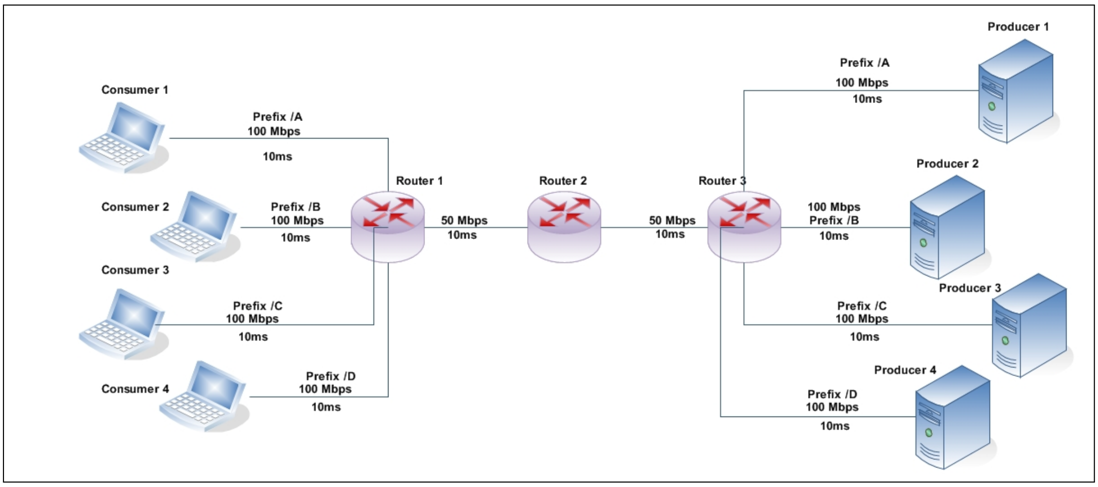


Figure 5.9. Dumbbell Topology

5.2.2 Dumbbell Topology

Dumbbell topology is used in many congestion network simulations [58], and several researchers [38, 42, 96, 145, 144, 143] have used it for performance evaluating of congestion control in NDN. The simulation parameters for the dumbbell topology in Figure 5.9 and Table 5.2

Table 5.2
Dumbbell Simulation Parameters

Parameter	Description
Number of Consumer	4
Number of producer	4
Number of Router	3
Number of links	10
Link Delay	10ms
Consumer Link Bandwidth to Router	100Mbit/s
Producer to Router	100Mbit/s
Between Routers	50Mbit/s
Simulation Time	100s

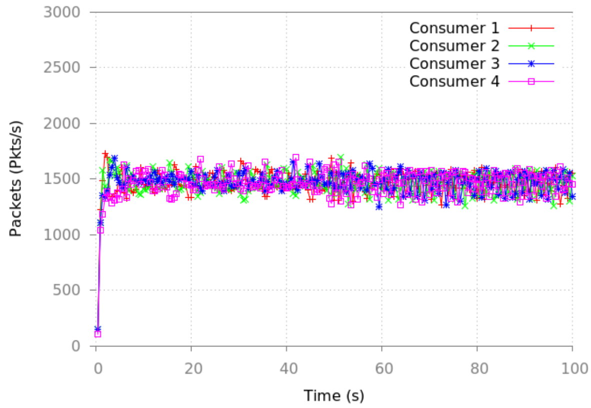
5.2.2.1 Equal Packet Size Scenario

The throughput is measured for individual consumers for HRCM, PCON and HIS, as illustrated in Figures 5.10 (a), 5.10 (b) and 5.10 (c) respectively. In Figure 5.10 (a) the throughput of four consumers is measured to indicate performance in HRCM.

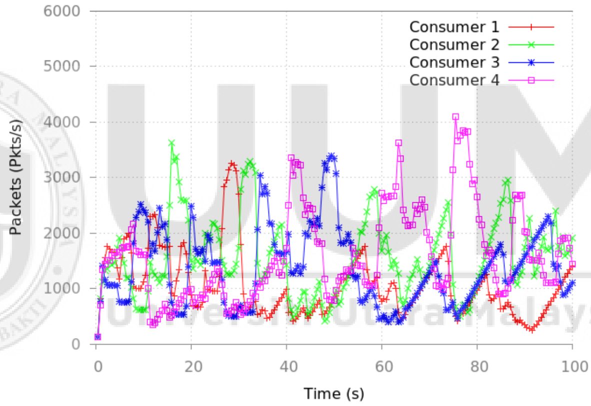
Consumers with 100 Mbps share the bottleneck link of 50 Mbps, resulting in serious congestion in attempting to accommodate four consumers at the same time. The result shows how HRCM shared the link between the four consumers with a stable allocation within the range of 1400 to 1600 packets throughout the simulation. Briefly, the stability of HRCM is 200 packets across the total simulation time for all consumers. Figure 5.10 (b) shows how PCON shared the bottleneck link between the consumers. The allocation looks unstable with a wide range across the simulation time and consumers, indicating a range of 800 to 3200 packets across the simulation time. At one time consumers 4 and 2 have the highest packet allocation, nearly reaching 4000. Hence, the stability of PCON can be represented on average as 2400 packets. Lastly, Figure 5.10 (c) shows the HIS throughput of the four consumers, managing the link with an average range of 1250 to 1750 packets for all consumers across the simulation time. The stability of HIS is encouraging with 500 packets. Comparing the performance of HRCM against PCON and HIS, the overall result shows that HRCM performs better with the difference of 200 packets, which is lower than 500 packets for HIS and 2400 for PCON. Hence HRCM is 60% better than HIS and 93% better than PCON, the worst in terms of link stability.

Figure 5.11 represents packet delay as the time taken for each consumer to send an Interest and receive a Data packet. HRCM shows the delay across the four consumers as between 0.08 and 0.09 in Figure 5.11 (a); PCON is between 0.08 and 0.1 in Figure 5.11 (b); and HIS is between 0.1 and 0.2 in Figure 5.11 (c). HRCM thus has the minimum delay and performs 6% and 32% better than PCON and HIS, respectively.

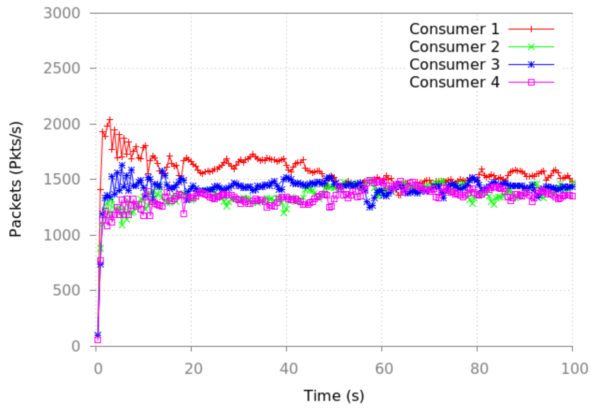
Link utilization and fairness of the bottleneck for HRCM, PCON and HIS are measured and presented in Figures 5.12 and 5.13. HRCM and HIS give similar performances, with a bandwidth utilization rate of 49Mbps for HRCM and 47Mbps for HIS.



(a) HRCM

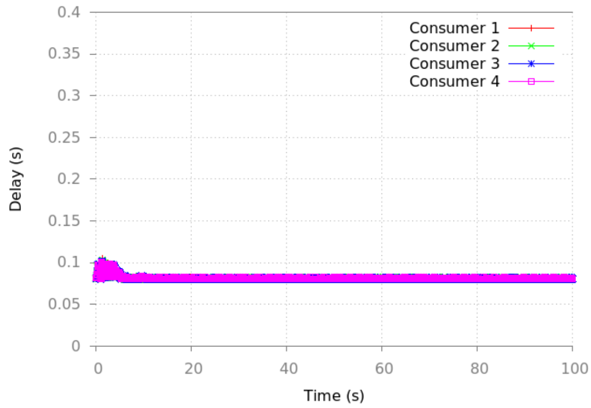


(b) PCON

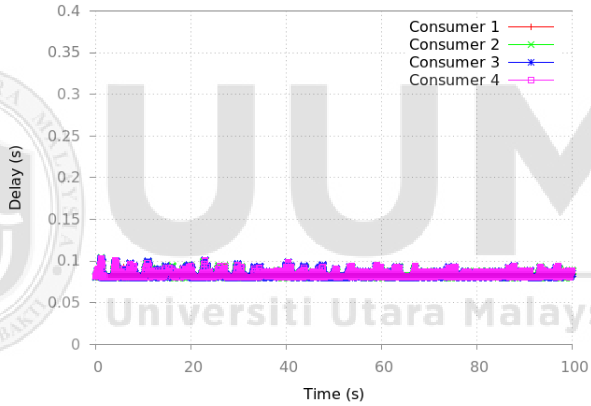


(c) HIS

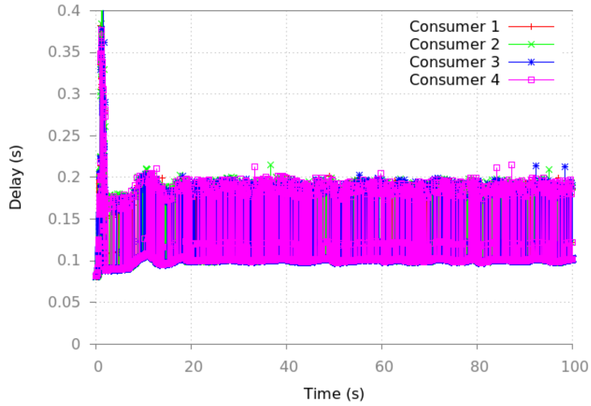
Figure 5.10. Throughput



(a) HRCM



(b) PCON



(c) HIS

Figure 5.11. Delay

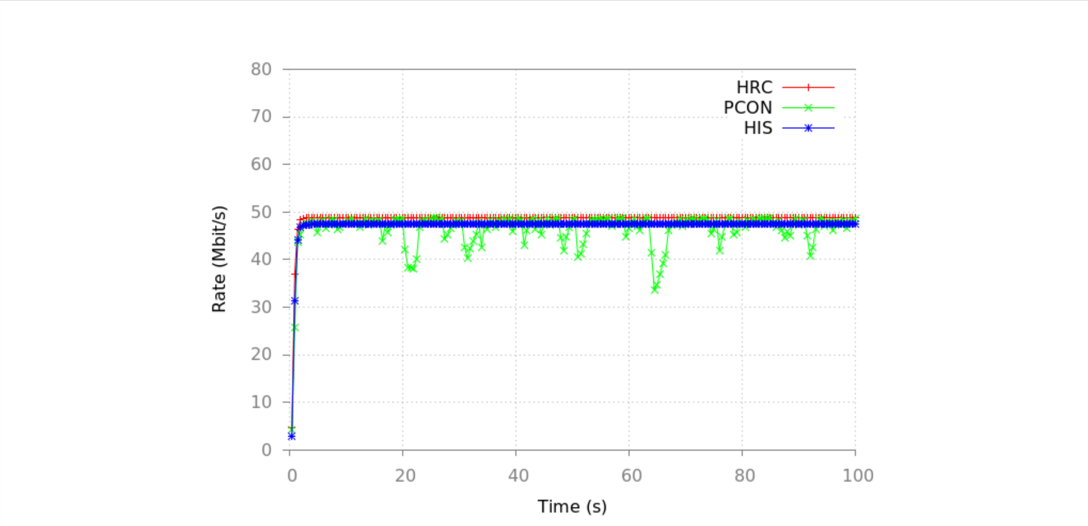


Figure 5.12. Link Utilization

This similarity is because both use shaping control by delaying the packets when there is congestion until the link is cleared. The PCON bandwidth utilization fluctuates between 38 and 47Mbps throughout the simulation time. PCON became unstable due to the lack of any shaping mechanism inside the router; it also utilizes the router only to detect the congestion, resulting in its poorer performance. In the case of fairness shown in Figure 5.13, PCON remains much more unstable than in the bandwidth utilization. It keeps fluctuating between 60% and 100% while HIS and HRCM are more stable. From the beginning of simulation time to 20 sec, HRCM performed better than HIS, although suddenly both attained closer performance, even though HRCM is 11% better overall than HIS. Both HIS and HRCM are better than PCON for bandwidth utilization with a high or perfect difference. In short, the performance of HRCM is better than PCON with a 60% improvement, and 11% improvement over HIS.

5.2.2.2 Different Packet Size Scenario

This section describes and compares the performance of HRCM, PCON and HIS in the scenario with different packet sizes delivered, and their link utilization. The result, presented in Figure 5.14, shows the number of packets delivered against simulation time and packet rate against simulation time for HRCM in Figure 5.14 (a) and (b),

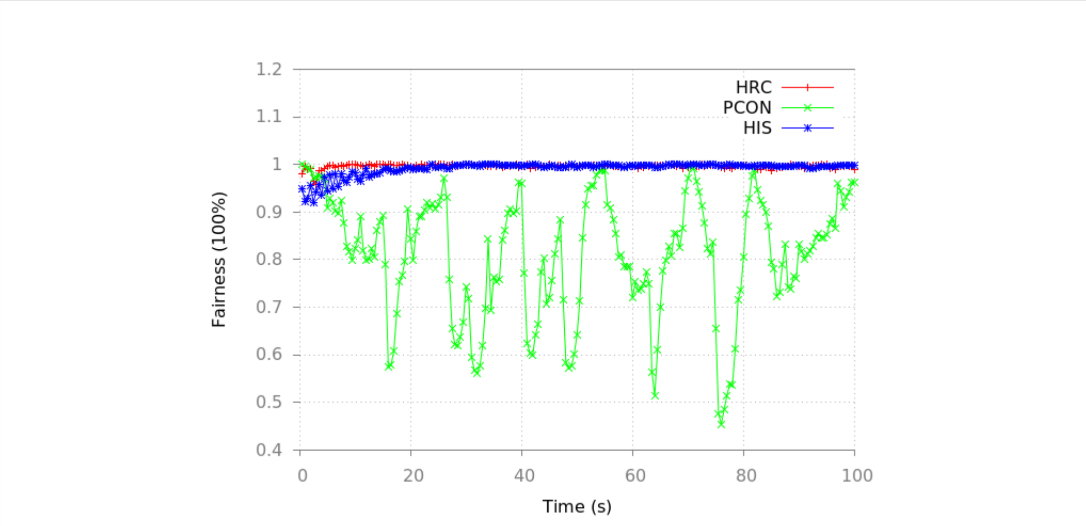
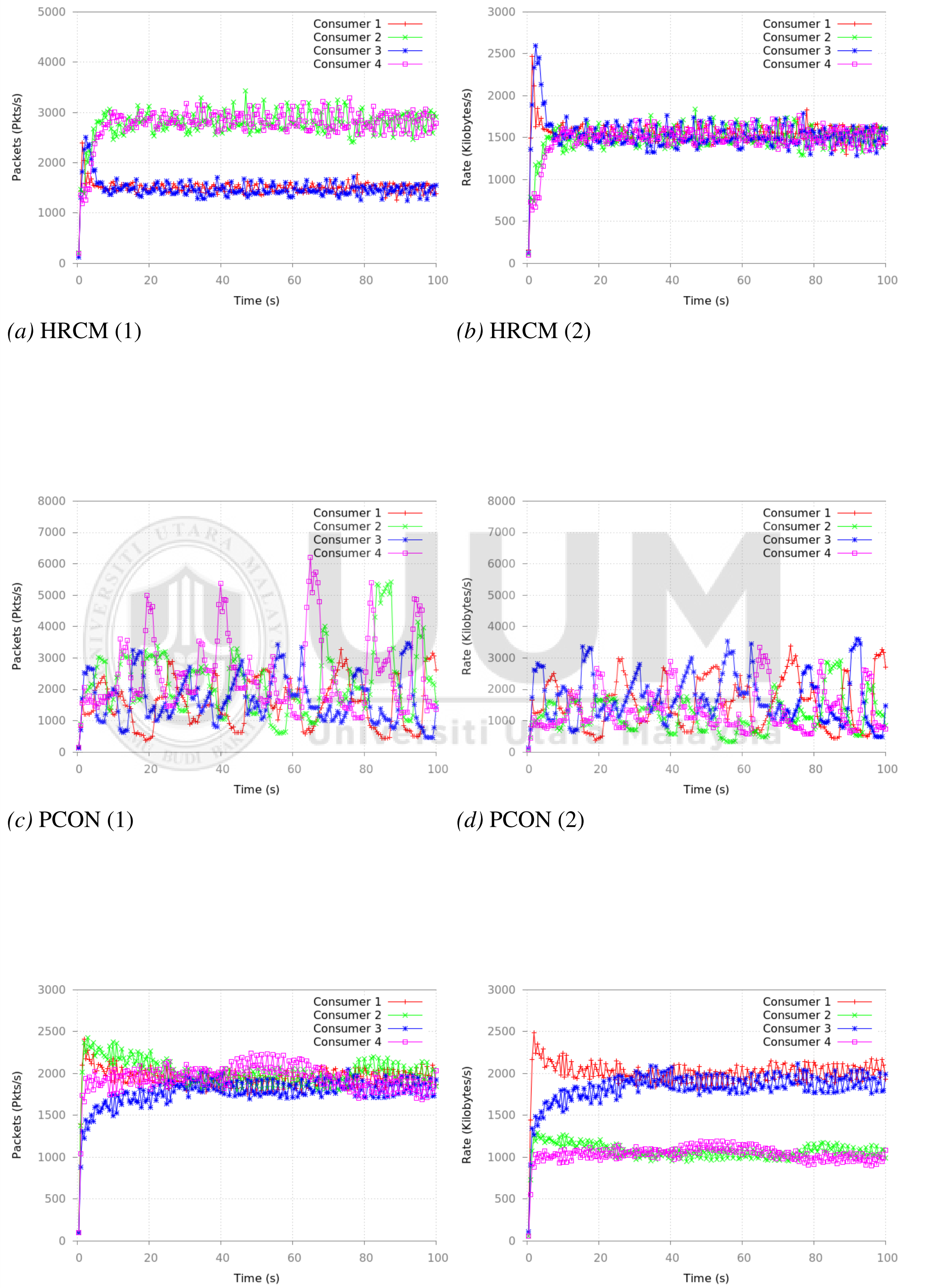


Figure 5.13. Fairness

PCON in Figure 5.14 (c) and (d), and HIS in Figure 5.14 (e) and (f). The simulation is run by assigning different packet rates for two sets of consumers; that is, consumers 1 and 3 are assigned 1024 bytes and consumers 2 and 4 512 bytes, to monitor how the links are utilized. For HRCM, the result in Figure 5.14 (a) shows stable rates for all consumers from the beginning to the end of the simulation. The consumers with 512-byte packets maintained the range closer to 3000 packets throughput. The same pattern is shown for the consumers with 1024-byte packets. Figure 5.14 (b) shows how HRCM manages the link with fairness among the four consumers. From the beginning of the simulation time consumers 1 and 3 reach as high as 2500 because of their packet size (1024 bytes), and consumers 2 and 4 reach only the 1500 rate. The HRCM reduces the highest startup of consumers 1 and 3 to stimulate the fair link utilization with consumers 2 and 4, meaning that it considers the number of consumers accessing the link and allocates the bandwidth without considering their packet rates.

Figures 5.14 (c) and (d) show the performance of PCON with the same scenario. Despite the different packet size allocation to consumers, their rate is not as stable as HRCM, as shown in Figure 5.14 (c). Also, in Figure 5.14 (d), the consumers' link utilization has a similar pattern to that in Figure 5.14 (c). HIS, presented in Figures



(a) HRCM (1) *(b)* HRCM (2)

(c) PCON (1) *(d)* PCON (2)

(e) HIS (1) *(f)* HIS (2)

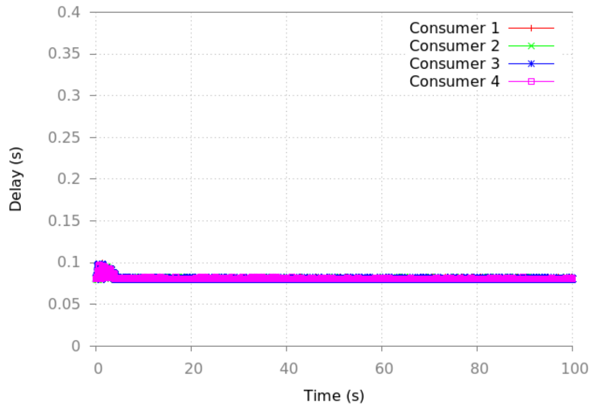
Figure 5.14. Dumbbell Throughput

5.14 (e) and (f) is the opposite of HRCM. For the link utilization, the pattern of HIS is most similar to that of HRCM shown in Figure 5.14 (a), the only difference being at the starting point where HIS shows consumers competing to take advantage of the link utilization.

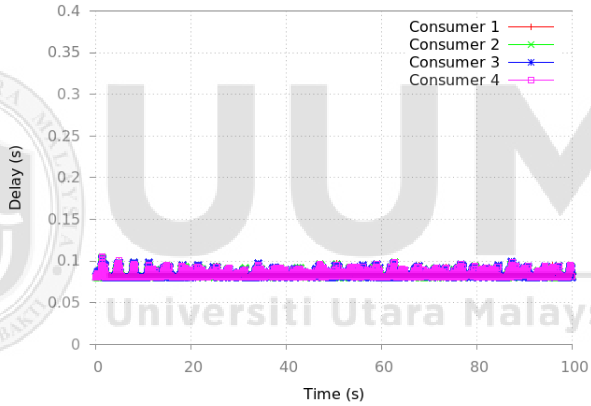
The delay is also observed with the different packet sizes allocated to consumers 1 and 3, as well as consumers 2 and 4, as explained. The result is similar or almost equal to that presented in Figure 5.11, with little difference for HIS. Figure 5.15 represents the consumers' packet delay as the time taken for each consumer to send an Interest and receive a Data packet. HRCM shows the delay across the four consumers is between 0.08 and 0.09 in Figure 5.15 (a); PCON is between 0.08 and 0.11 in Figure 5.15 (b); and HIS is between 0.1 and 0.23 in Figure 5.15 (c). By comparison, HRCM has a minimum delay and performs better by 10% and 48% than PCON and HIS, respectively.

Different packet sizes are assigned to each mechanism to determine and compare their performance. Link utilization and fairness are observed with respect to the different packet size, as shown in Figures 5.16 and 5.17. The performance of HRCM, PCON and HIS are similar to the results presented in Figure 5.12 for link utilization. Both HRCM and HIS have the almost same performance, while PCON remains unstable. Although the results in Figure 5.16 are similar to those in Figure 5.12 for link utilization, Figure 5.17 is totally different from Figure 5.13 in terms of fairness. As shown in Figure 5.17, the superior performance of HRCM is obvious, while PCON retains its unstable behaviour and HIS remains uniform.

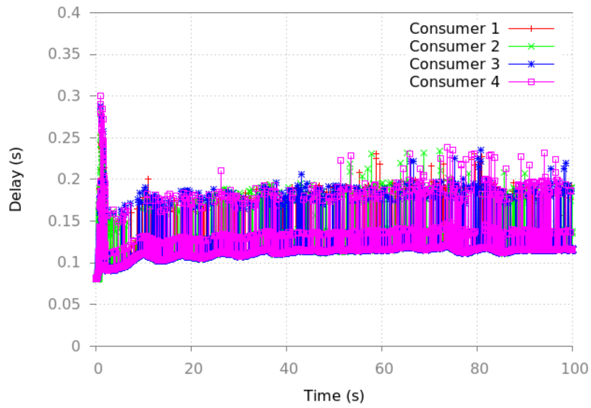
Numerically, the performance of HRCM is improved by 10% over HIS and 20% over PCON, which is far better than the link utilization presented in Figure 5.12 where there



(a) HRCM



(b) PCON



(c) HIS

Figure 5.15. Delay

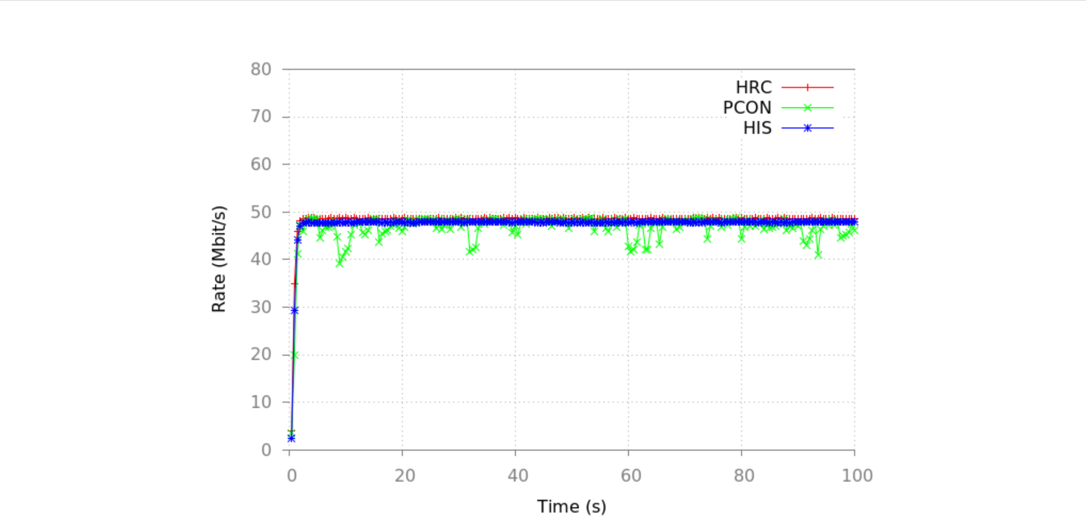


Figure 5.16. Link Utilization

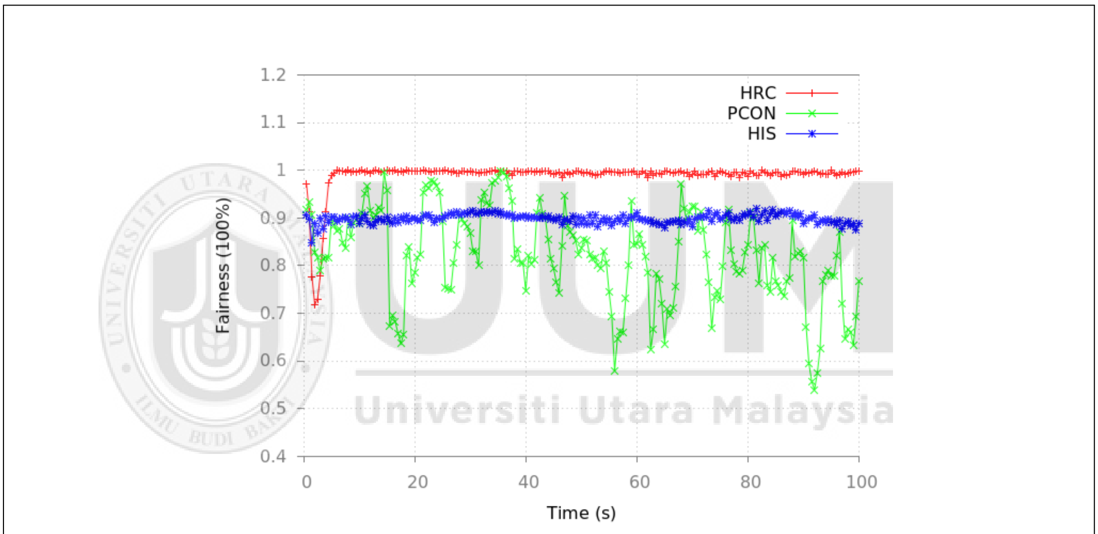
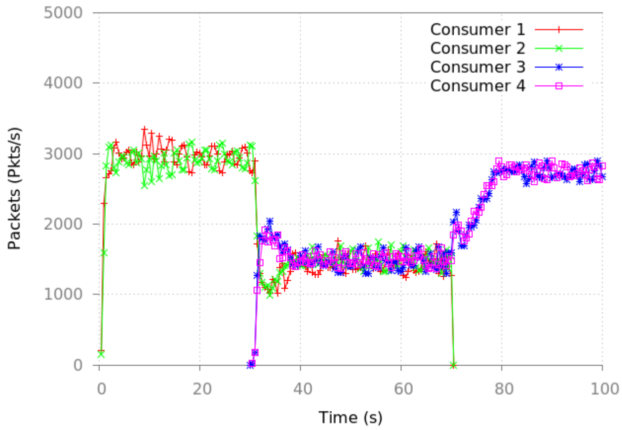


Figure 5.17. Fairness

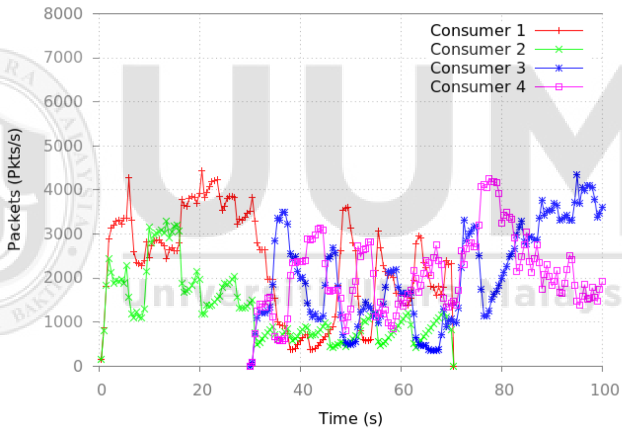
was no clear improvement against HIS. HRCM performed well as a result of per bit scheduling and fairness between the consumers, allocating equal bandwidth between the consumers with clear fairness.

5.2.2.3 Different Starting Time Scenario

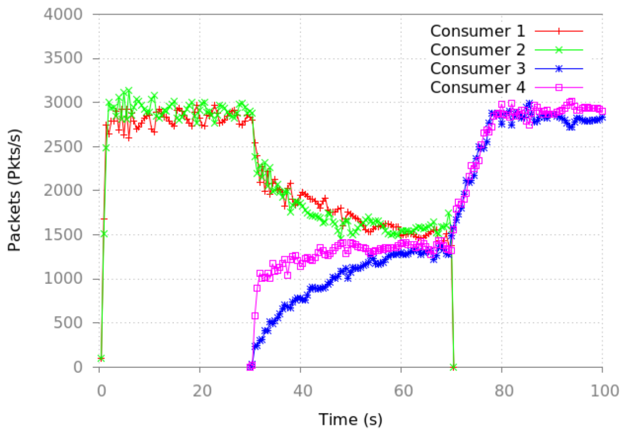
Different starting times are applied in this scenario to determine comparative adaptability. At the beginning of the simulation two consumers start at the same time 1 and 2 from 0 sec and stopping at 70 sec; consumers 3 and 4 start at 30 sec and stop at the end



(a) HRCM



(b) PCON

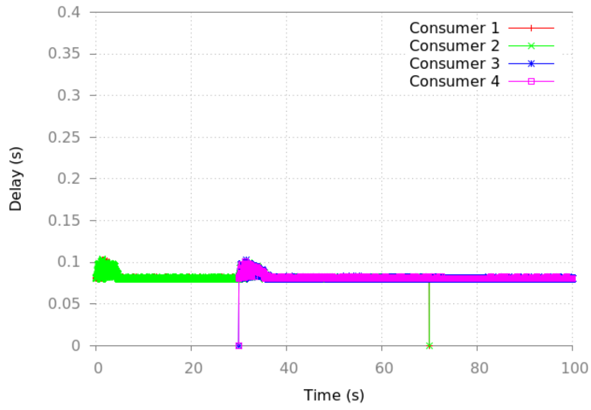


(c) HIS

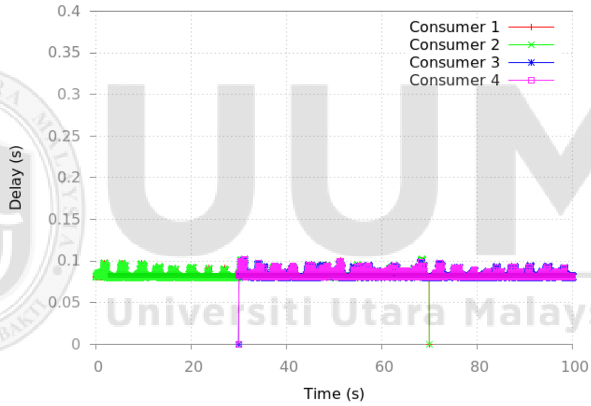
Figure 5.18. Throughput

of the simulation time, that is 100 sec. The throughput of HRCM is highly adaptable due to its fair scheduling of access for different consumers despite their packet size. Fairness is giving due consideration among the consumers with different packets sizes and different starting times. From Figure 5.18 (a), which highlights the performance of HRCM, it is shown that within only 5 sec the mechanism adapts the incoming packets from consumers 3 and 4 which started 30 sec into the simulation time. In Figure 5.18 (b), the performance of PCON is similar to that of HRCM, although the adaptability started 50 sec into the simulation time, which means that PCON adapted within 20 sec and the adaptation was not fair among the four consumers. The bandwidth allocation is dispersed within the range of approximately 800 to 4000 packets per sec. The performance of HIS is presented in Figure 5.18 (c) with the same settings as HRCM and PCON. The adaptability is spotted 60 sec into the simulation time, that is when four consumers have nearly equal treatment, although consumers 1 and 2 nearly reached their stopping time; in short, HIS adapted within 30 sec. The performance of HRCM is 75% and 83% better than PCON and HIS, respectively. In short, HRCM is promising in terms of adaptation even with different start-up times between consumers.

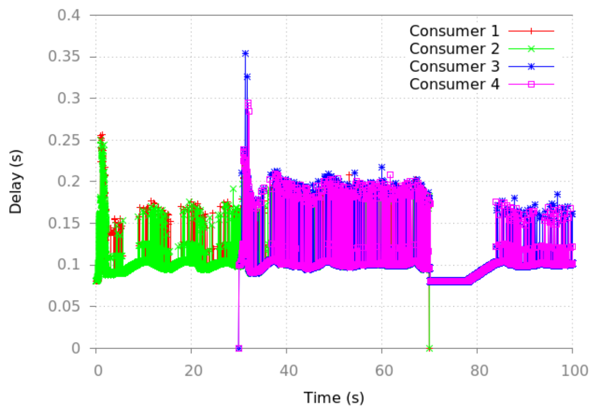
The delay is also observed with different starting times for consumers 1 and 2, as well as consumers 3 and 4, as discussed. The delays for HRCM, PCON and HIS have some similarities, with delay accounted for by the different time allocated to consumers, apart from certain points that are highlighted in Figures 5.19 (a) and 5.19 (c) for HRCM and HIS. PCON maintained its delay at 0.08 to 0.11, as shown in Figure 5.19 (b), while for HRCM there is sharp stability of the delay at 73 sec of simulation time to around 83 sec. This coincided with the end time of consumers 1 and 2, that is, the links have extra bandwidth released by those consumers. Similarly, in HIS, the same sharp stability started at 70 sec to 80 sec of simulation time. In comparison, HRCM and HIS have common functionality, which is the sign of adaptability as



(a) HRCM



(b) PCON



(c) HIS

Figure 5.19. Delay

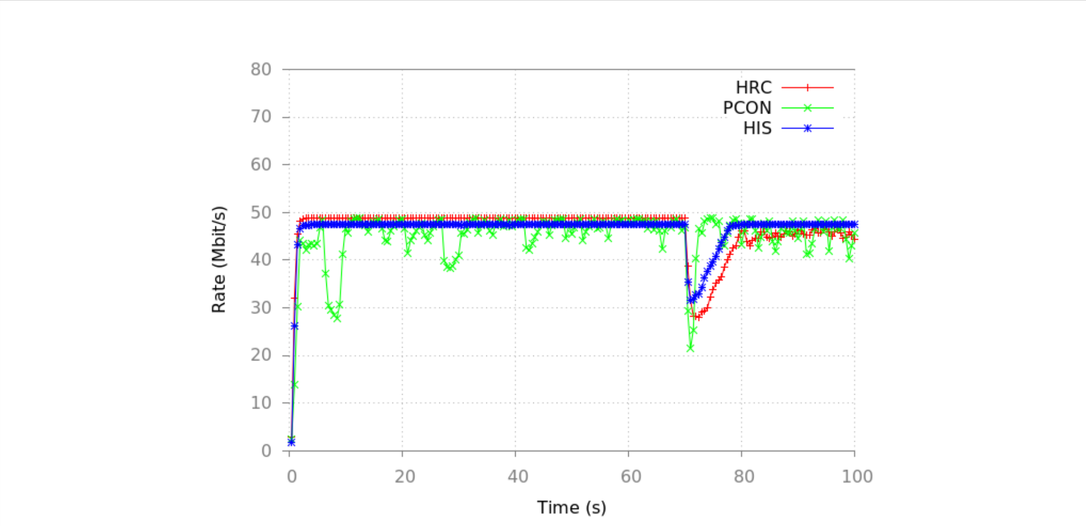


Figure 5.20. Link Utilization

presented in Figure 5.18 (a) and (c). The over-performance of HRCM is almost the same as the results presented in Figures 5.11 and 5.15, with only a difference of 10 sec stability around 70 sec of simulation time.

In this scenario, the link utilization and fairness are observed to measure performance when pairs of consumers started and ended at different times. This is in line with the adaptive behaviour of HRCM tested against PCON and HIS. Figure 5.20 highlights the link utilization when consumers 1 and 2 start at 0 sec and stop at 70 sec, while consumers 3 and 4 start at 30 sec and stop at 100 sec. The same conditions are met in Figure 5.21, which compares the fairness results. The link utilization result shows a sudden declination of the rate for all three mechanisms at 70 sec to 80 sec of the simulation time. HRCM and HIS have a similar pattern with a slightly better rate for HRCM, while PCON still shows its normal instability as described in earlier results. Figure 5.21 shows the fairness and stability of link utilization for the three mechanisms. HRCM and HIS are very stable with the access of the two consumers who started at the beginning of simulation time, while PCON continues to fluctuate. At 30 sec simulation time, the two traditional consumers started accessing the link, and this is what changes the rate of all mechanisms. The changes suddenly increase the rate of

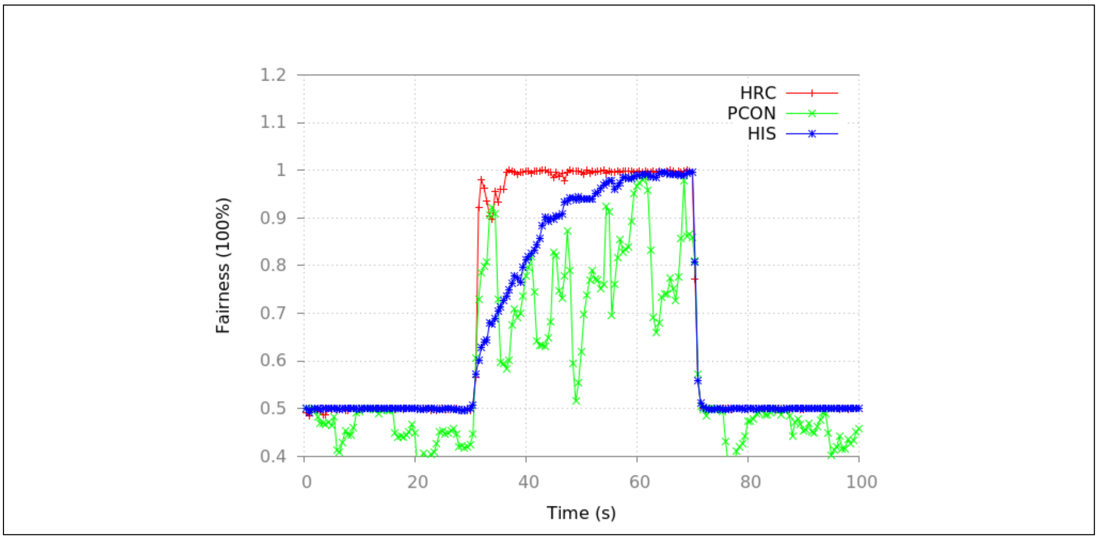
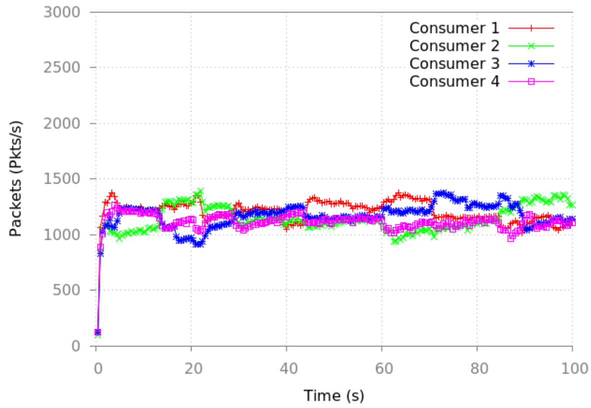


Figure 5.21. Fairness

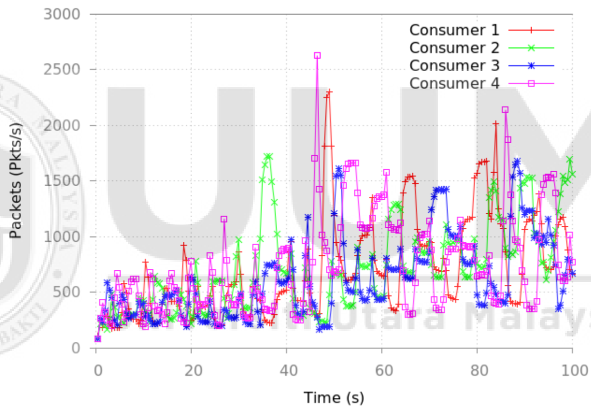
HRCM to 100%; the rate of HIS is inclining and reaches stability at 60 sec of simulation time where it coincides with HRCM. PCON fluctuates within the range without specific or fair action. In general, HRCM quickly adapts to any changes, while HIS takes around 30 sec slowly to reach the adaption level and PCON has no adaptability.

5.2.2.4 Multipath Scenario

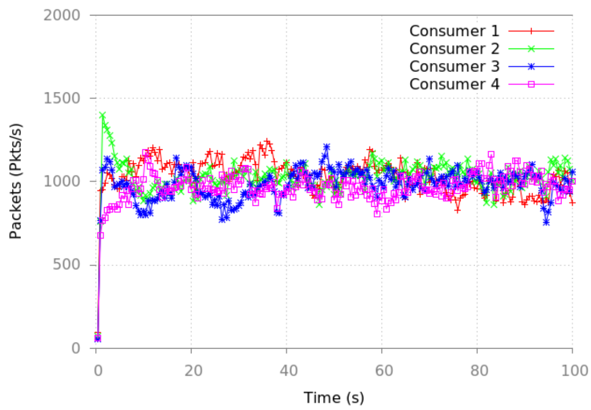
The throughput of Dumbbell topology with four consumers with the same size of packet is measured, while changing the links from router to producers by 10Mbit/s and install all prefix in each producer. The results shown in Figure 5.22 (a), (b) and (c) for HRCM, PCON and HIS, respectively. HRCM adapted the transmission of the packets from the starting point. It tried to adapt and give fair access to each consumer; as shown in Figure 5.22 (a) the consumers maintained the number packets between 1000 and 1300 throughout the simulation time. PCON, as shown in Figure 5.22 (b), utilizes a single path until other available paths are discovered, and then tries to share them. This is what makes PCON become unstable in terms of utilization and fairness of link access. In Figure 5.22 (b), at the point of around 45 sec of simulation time, the link distribution between four consumers is highly unfair; consumers 1 and 4 reach the highest level of up to 2500 packets, while consumers 2 and 3 fall to 200 packets. The



(a) HRCM



(b) PCON



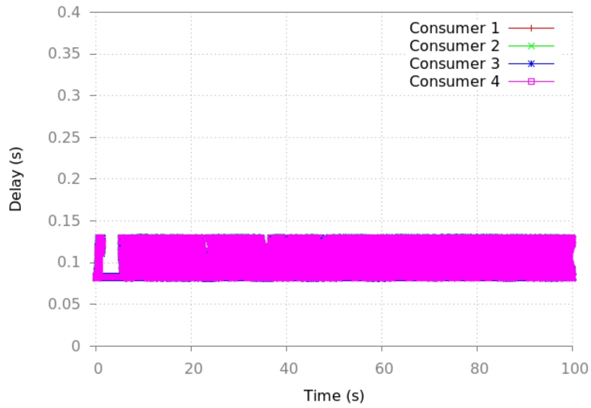
(c) HIS

Figure 5.22. Dumbbell Throughput

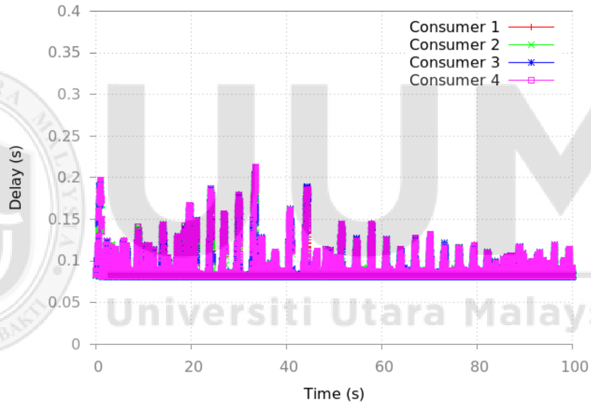
performance of HIS is somewhat similar in the case of adaptability, but the fairness of link utilization is not the same as HRCM; see Figure 5.22 (c).

The delay measured for the dumbbell topology shown in Figure 5.23 is a graph of packet delay in sec against simulation time. The result is presented in three periods of simulation time: 0 to 10 sec, 10 to 60 sec and 60 to 100 sec. In the 0 to 10 sec period, the mechanisms adapt to the forwarding between all paths; HRCM's performance is higher than that of HIS by 63%, and 25% against PCON, as their packet delays are unstable. At 10 to 20 sec of simulation time, the performance of HRCM and HIS becomes stable at 0.08 to 0.13 and 0.8 to 0.28 sec of packet delay respectively, whereas PCON keeps on fluctuating between 0.08 to 0.23 sec. For the 60 to 100 sec period, HRCM and HIS maintain the delay as in the previous situation, while PCON becomes stable between 0.08 and 0.13. Overall performance for the complete simulation time shows that HRCM has minimal packet delay compared to PCON by 25% until PCON takes over all the available paths; HIS is 41%. The worst performance, HIS, occurs because of the queue length policy to determine congestion: while the queue lengthens the packet delay will increase. Both HRCM and PCON use the queue packet delay policy to determine the congestion, but unlike HIS notification of congestion is sent to the consumer as long as the queue is increasing, without waiting until the queue is full. Conclusively, HRCM is better than PCON, especially between 10 to 100 sec of simulation time, because HRCM uses all the available path from the beginning of the forwarding, unlike PCON which uses only the best path before changing to multipath.

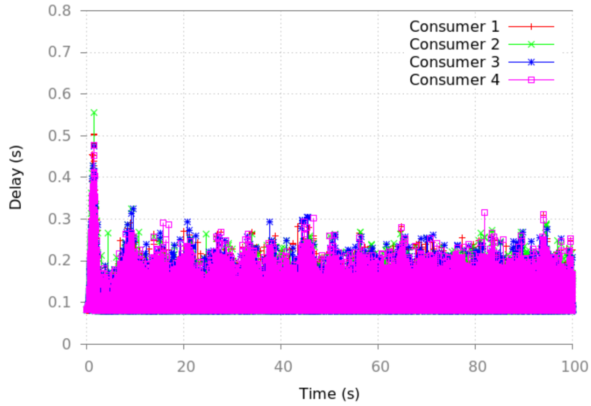
The multiple path scenario is run for HRCM, PCON and HIS for 100 sec and the results for link utilization and fairness are presented in Figures 5.24 and 5.25. Link utilization is plotted on a graph of rate in Mbps against the simulation time. From the beginning of the simulation HRCM shows better performance, reaching 38Mbits/s;



(a) HRCM



(b) PCON



(c) HIS

Figure 5.23. Delay

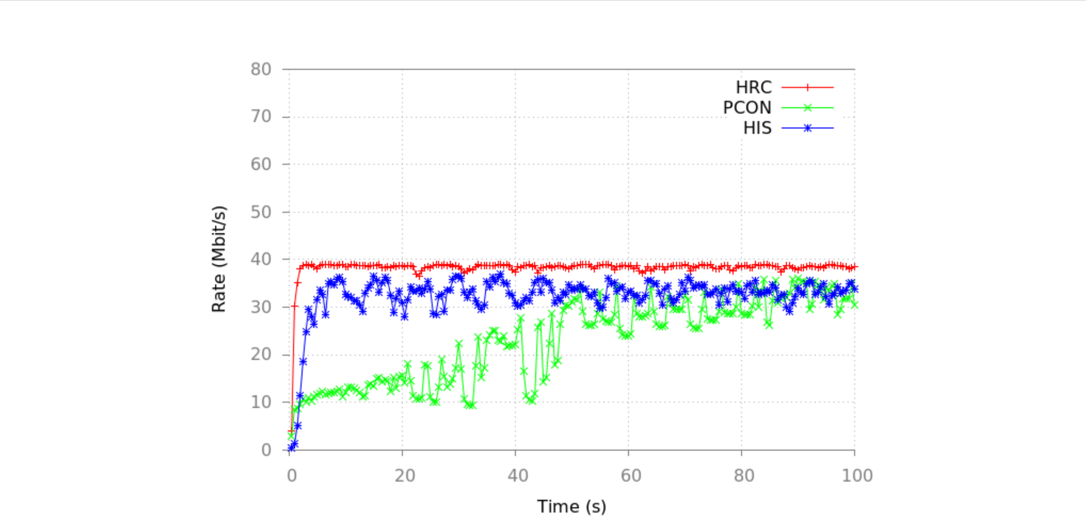


Figure 5.24. Link Utilization

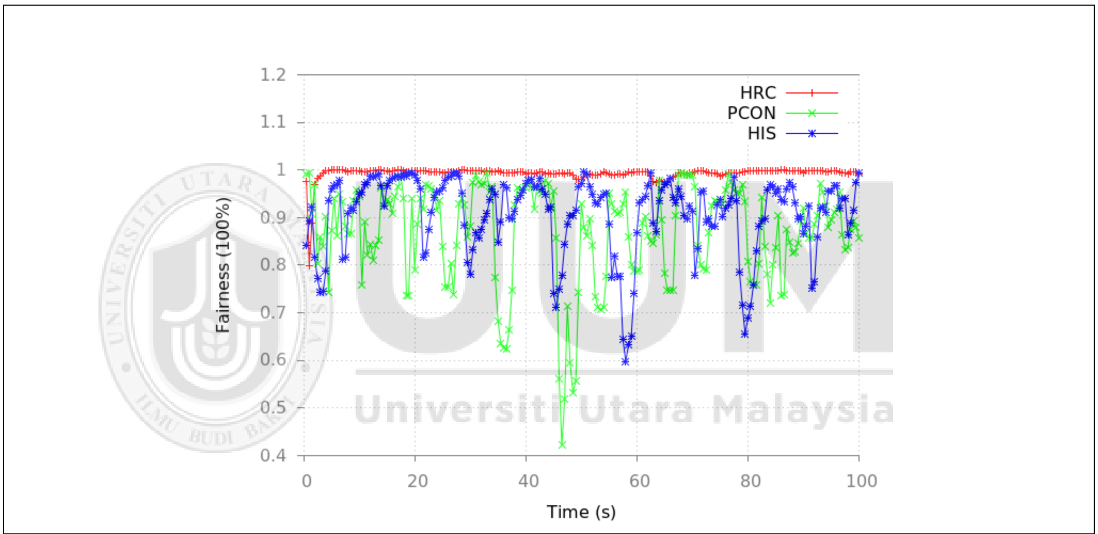


Figure 5.25. Fairness

HIS followed from 5 sec of simulation time and reached 33Mbits/s and lastly PCON reached 10Mbits/s, gradually surging to 30Mbits/s when the simulation time reached 60 sec. The changes in PCON have the same explanation as for Figure 5.22 (b) above. The multiple-path scenario confirmed the improvement in link utilization performance of HRCM over HIS and PCON by 13% and 21% respectively. Further, the fairness of a multiple path scenario is measured and presented in Figure 5.25 for all the mechanisms. HRCM maintains the fairness of link utilization throughout the simulation time with adequate stability at the rate of 100%, while PCON fluctuates within the range

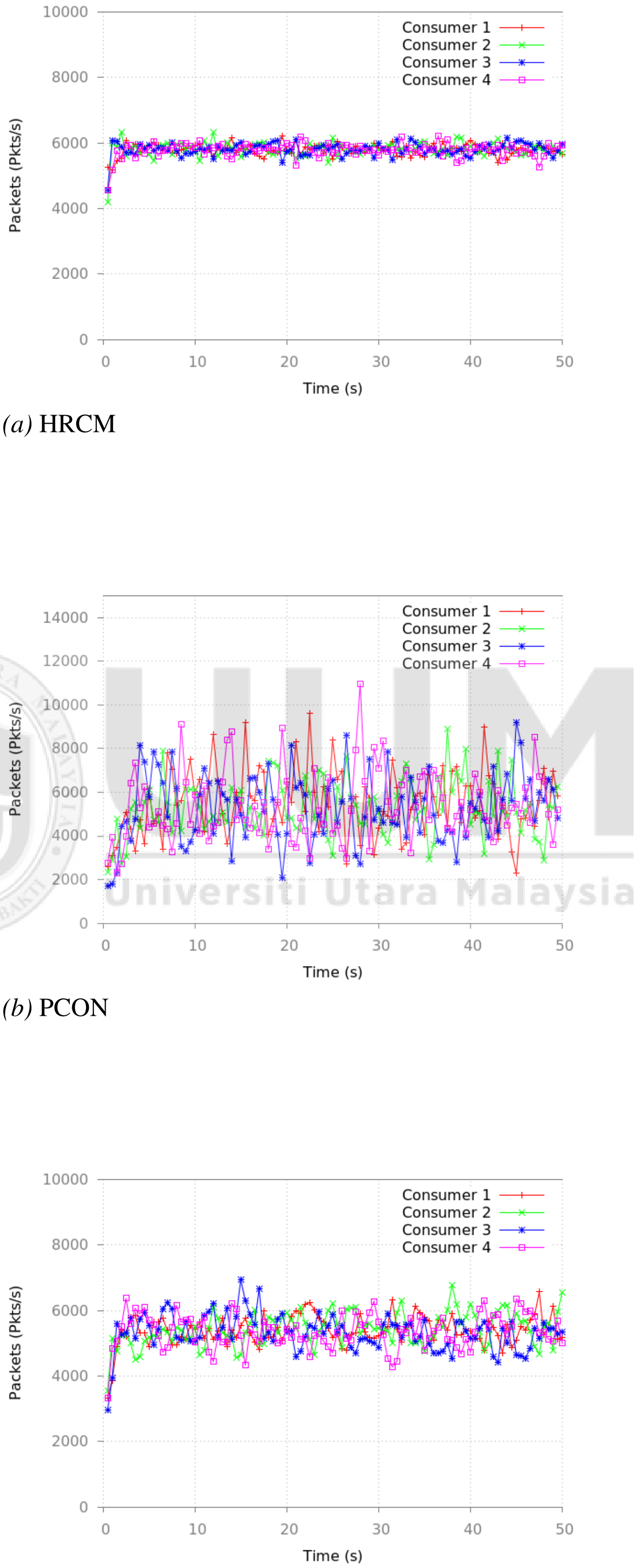
60 to 100%, although at around 50 sec simulation time, the rate falls to approximately 40%, as shown in Figure 5.25. In the same figure, the multiple path fairness of HIS is presented with average fairness of 90%. The inference from the overall observation of fairness is that HRCM performs better than PCON with 20% improvement, and better than HIS with 10%.

5.2.3 Abilene Typology

To evaluate the efficiency of the proposed mechanism in a real topology setting we conducted several simulation-based experiments in various scenarios using Abilene topology. This was created by the Internet2 community and connects regional network aggregation points to provide advanced network capabilities to over 230 Internet2 institutions in the US. Recent studies have stressed the importance of using the Abilene topology especially for the probing result introduced in [142, 143]. The Abilene topology in this study (see Figure 3.10 and Table 5.3) consists of eleven NDN routers, four consumer nodes, eight producer nodes, and links 26. It applies all the default routing, connection, and delay settings of Abilene except for using 1% of link bandwidth to create congestion and a bottleneck in this topology.

Table 5.3
Abilene Simulation Parameters

Parameter	Description
Number of Consumer	4
Number of producer	8
Number of Router	11
Number of links	26
Link Delay	Default
Consumer Link Bandwidth to Router	100Mbit/s
Producer to Router	100Mbit/s
Between Routers	100Mbit/s
Simulation Time	100s



(a) HRCM

(b) PCON

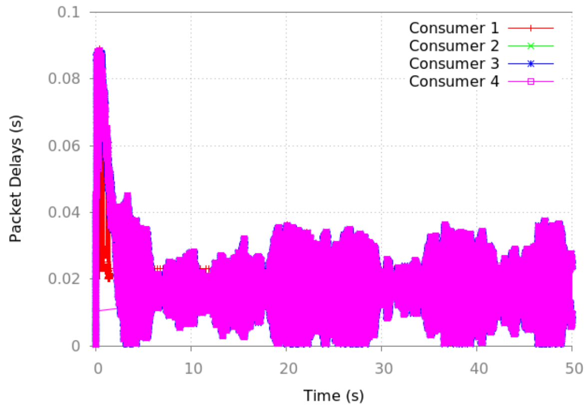
(c) HIS

Figure 5.26. Throughput

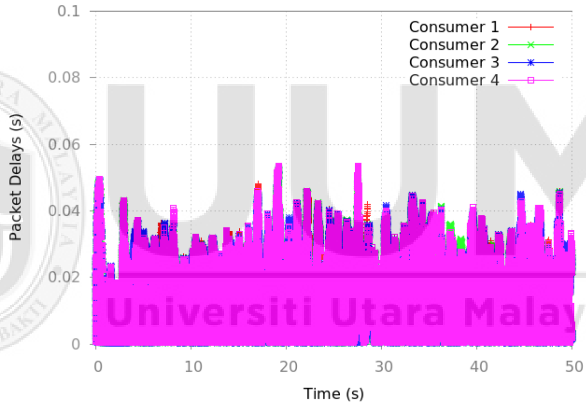
5.2.3.1 Equal Packet Size Scenario

The throughput is measured for the individual consumers in the Abilene topology with the same packet size scenario for HRCM, HIS and PCON, as illustrated in Figure 5.26 (a) 5.26 (b) and 5.26 (c) respectively. The link must result in high congestion in accommodating four consumers at once, reflected in the performance of the three schemes. The result shows how HRCM shared the link among the four consumers with a stable allocation within the range of approximately 5200 to 6200 packets throughout the simulation. Figure 5.26 (b) shows how PCON shared the link; the allocation looks unstable with a wide range across the simulation time, from 2000 to 9000 packets. Hence, the instability of PCON is great enough to destabilize its performance with the difference of around 7000 packets. Finally, HIS throughput is shown in Figure 5.26 (c), managing the link with an average range of 4200 to 6200 packets for all consumers across the simulation time. The stability of HIS is encouraging with 2000 packets compared to PCON. The conclusive result of throughput for the Abilene topology revealed that HRCM still performs better by 50% against HIS and 85% against PCON, the worst in terms of link stability.

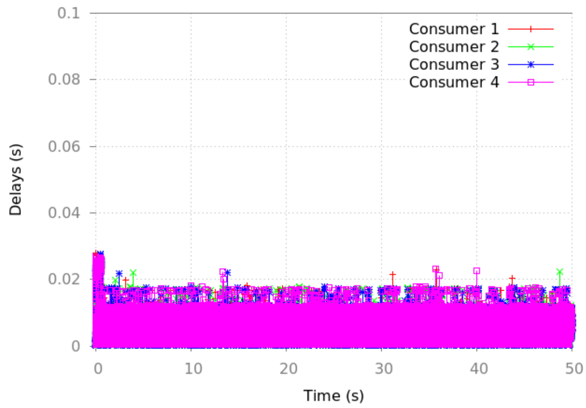
The results of delay in the Abilene topology are presented as a graph of packet delay in sec against simulation time. The observations of are recorded for two simulation periods, 0 to 3 sec and 30 to 50 sec. In the first period, the performance of HRCM is 40% higher than that for HIS, and 10% over PCON, as their packes delays are unstable. Between 3 and 50 sec of simulation time of the second situation, the performance of HRCM remains stable between 0.001sec and 0.03sec of packet delay; HIS stabilizes between 0.001 and 0.02, and PCON keeps on fluctuating between 0.001 and 0.004 sec. This shows only a moderate improvement in HRCM, which performs better by 12% than PCON, while HIS is the best of the three.



(a) HRCM



(b) PCON



(c) HIS

Figure 5.27. Delay

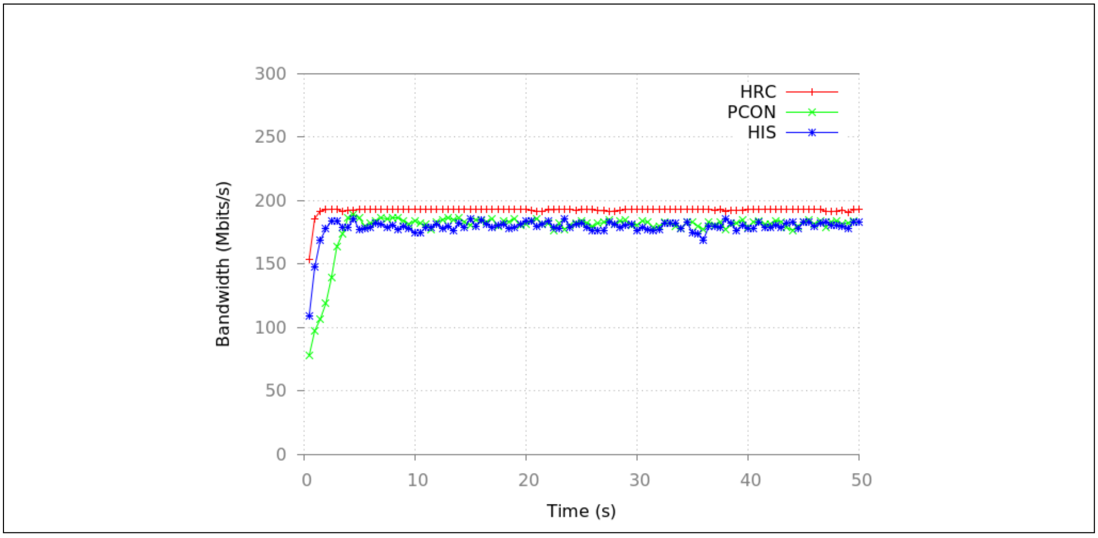


Figure 5.28. Link Utilization

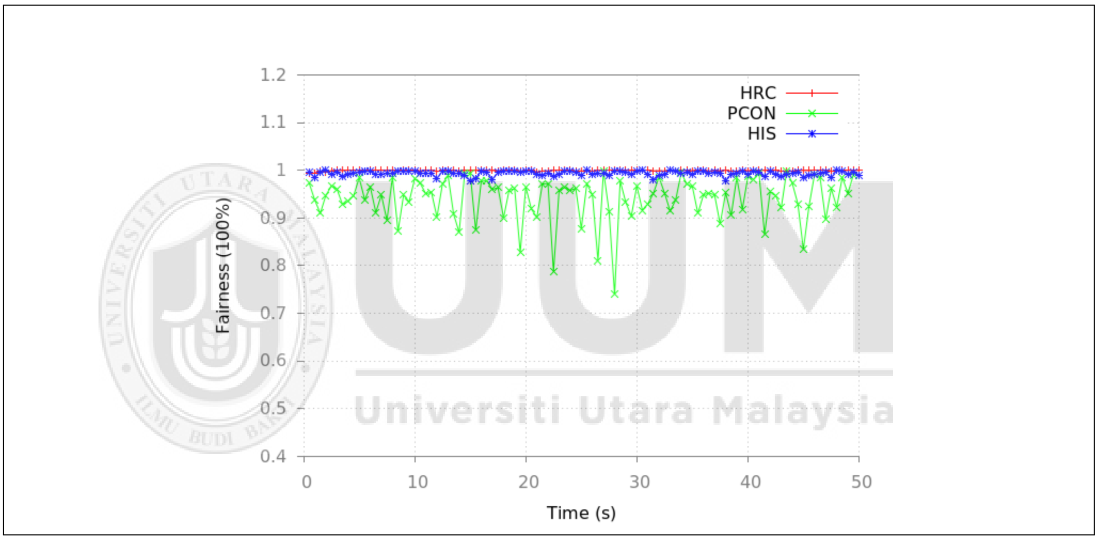


Figure 5.29. Fairness

The equal packets scenario under the Abilene topology is run to measure the link utilization and fairness, with comparisons presented in Figures 5.28 and 5.29. PCON and HIS give a similar performance, with a bandwidth utilization rate of 185Mbps for both. All mechanisms remained stable throughout the simulation time, unlike in the dumbbell topology. HRCM has stable bandwidth utilization at 194Mbps throughout the simulation time, better than PCON and HIS by only 5%. Their performance is almost the same because the nature of the Abilene topology means that they have different optimal paths between consumer and producer. In the case of fairness shown in Figure 5.29, PCON remains unstable, unlike the bandwidth utilization. It fluctuates between approximately 90 and 100%, while HIS and HRCM are more stable with an equal performance. Conclusively, the performance of HRCM and HIS are better than PCON with almost 10% improvement, for the reason given for the link utilization.

5.2.3.2 Different Packet Size Scenario

The three mechanisms were implemented in the Abilene topology to check their scheduling and fairness performance when the link handles different sized packets. Figure 5.30 indicates the number of packets delivered against simulation time and packet rate against simulation time for HRCM in Figure 5.30 (a) and (b), PCON in Figure 5.30 (c) and (d), and HIS in Figure 5.30 (e) and (f). The simulation is run by assigning the same packet rate for two consumers: consumers 1 and 3 are assigned 1024 bytes, and consumers 2 and 4 512 bytes. For HRCM, the result in Figure 5.30 (a) shows the two different groups of consumers with different packets numbers and stable rates for each group from the beginning to the end of the simulation. The consumers with 512-byte packets maintained a throughput of 5800 to 6100 packets, while those with 1024-byte packets maintained the range of 10,000 to 11500 packets. Figure 5.30 (b) shows how HRCM manages the link with fairness among the groups of consumers with two different packet sizes. From the beginning to the end of the simulation time, there is negligible disparity between the groups.

Figure 5.30 (c) and (d) shows the performance of PCON with the same settings. The rate is not stable, as shown in Figure 5.30 (c) and 5.30 (d). The number of packets increases from around 10 to 20 sec of simulation time for consumers 2 and 4, and for consumers 1 and 3increases from 0 to 10 sec of simulation time. HIS, represented in Figure 5.30 (e) and (f), shows the opposite behaviour to HRCM, trying to utilize the nearest number of packets for all the consumers as shown in Figure 5.30 (e) and using

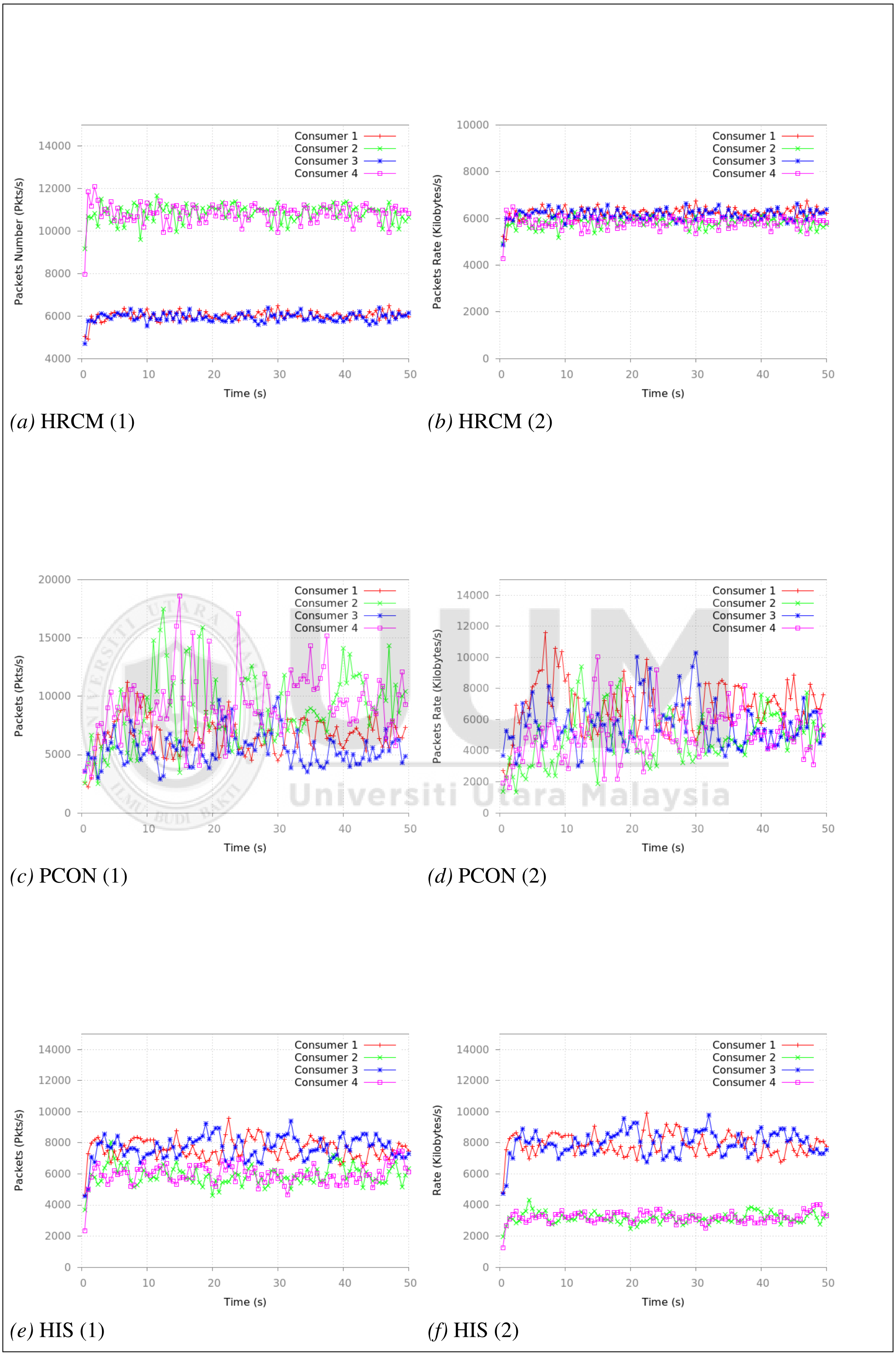


Figure 5.30. Throughput

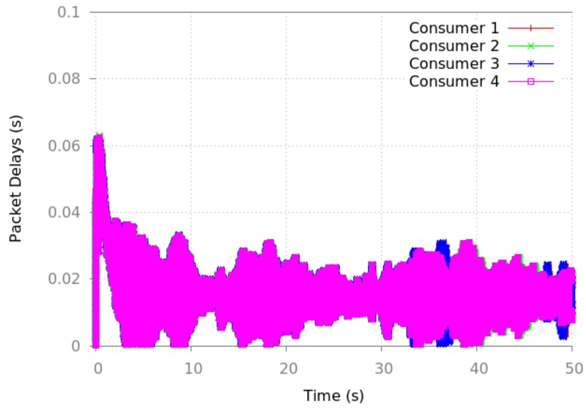
a different rate as shown in Figure 5.30 with consumers 2 and 4 maintaining the rate of 3000 to 4000 Kbits/s.

The delay is also observed with different packet sizes allocated to consumers 1 and 3, and consumers 2 and 4. The results are almost equal to those presented in Figure 5.28, with little difference for PCON. Figure 5.31 presents the consumer's packet delay as the time taken for each consumer to send an Interest and receive a Data packet. HRCM shows the delay across the four consumers as between 0.001 and 0.03 in Figure 5.31 (a), PCON is between 0.001 and 0.03 in Figure 5.31 (b), and HIS is between 0.001 and 0.018 in Figure 5.31 (c). By comparison, the HIS has a lower minimum delay than HRCM or PCON.

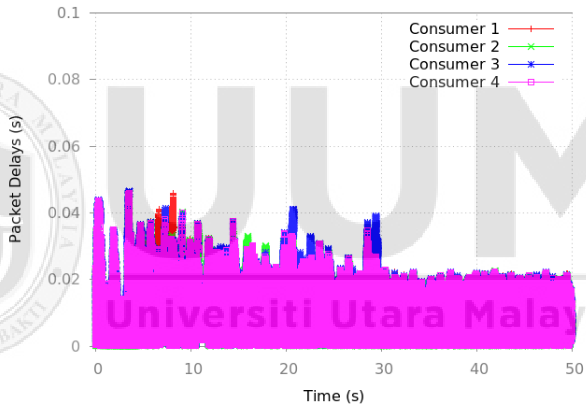
The results for link utilization and fairness are shown in Figures 5.32 and 5.33. The performance of HRCM, PCON and HIS is slightly different from the results presented in Figure 5.16 for the link utilization where PCON competes with HIS. Again, PCON and HIS are similar, while HRCM remains stable with higher link utilization. As shown in Figure 5.32 the performance of HRCM is visible, while the PCON and HIS maintain the same level of performance, 5% below HRCM. Figure 5.33 shows the fairness between four consumers, with HRCM performing well; fairness between the consumers accessing the link is maintained without considering their different packet size. That is, HRCM allocated equal bandwidth to the consumers with clear fairness, scoring 100%, against PCON's 90% and HIS's 80%.

5.2.3.3 Different Starting Time Scenario

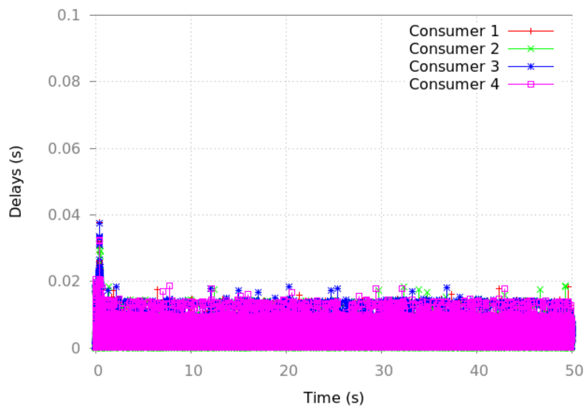
The Abilene topology with different starting times is applied in this scenario to compare the adaptability of HRCM, PCON and HIS. At the beginning of the simulation two consumers, 1 and 2, start at the same time, stopping at 35 sec; consumers 3 and 4



(a) HRCM



(b) PCON



(c) HIS

Figure 5.31. Delay

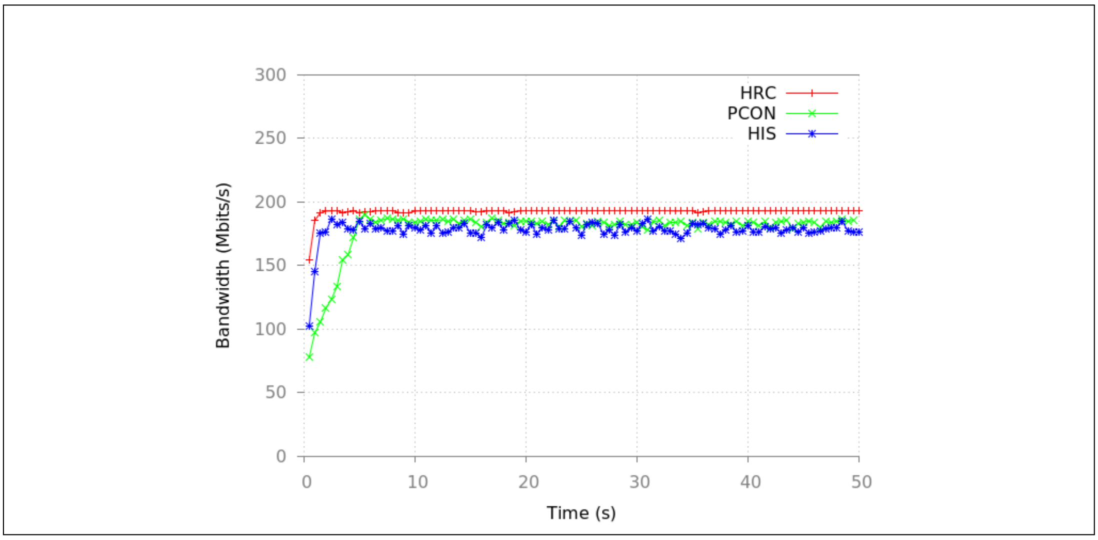


Figure 5.32. Link Utilization

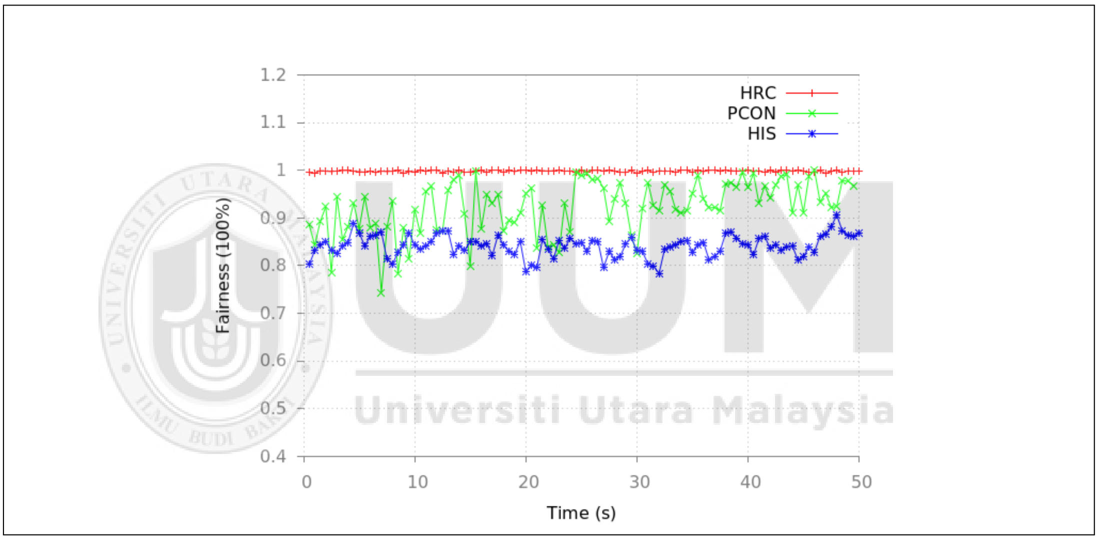
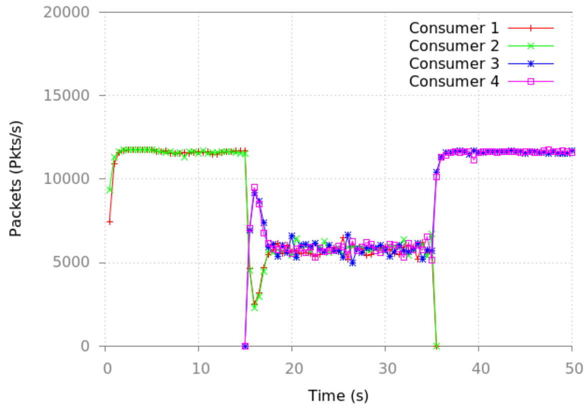
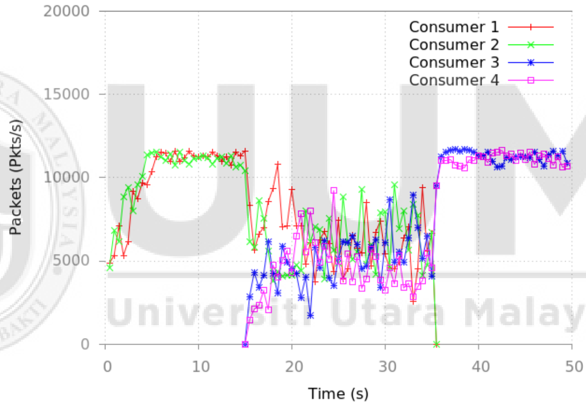


Figure 5.33. Fairness

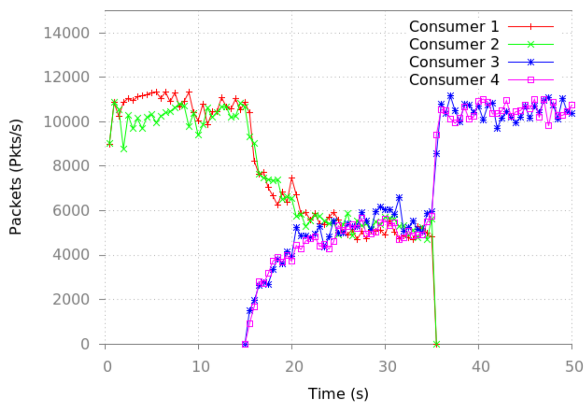
start at 15 sec and stop at the end of the simulation, 50 sec. Figure 5.34 (a) shows the performance of HRCM; at startup, the mechanism adapts the incoming packets from consumers 3 and 4 that started 15 sec into the simulation time. The number of packets stabilized at 12000 for consumers 1 and 2, immediately changing to 7000 with the incoming packets from consumers 3 and 4, for fair sharing of the link. Immediately after consumers 1 and 2 stop, the packets return to the normal number, 12000. The throughput of HRCM is thus highly adaptable due to fair scheduling with access from different consumers despite their packet sizes. That is, fairness gives due considera-



(a) HRCM



(b) PCON



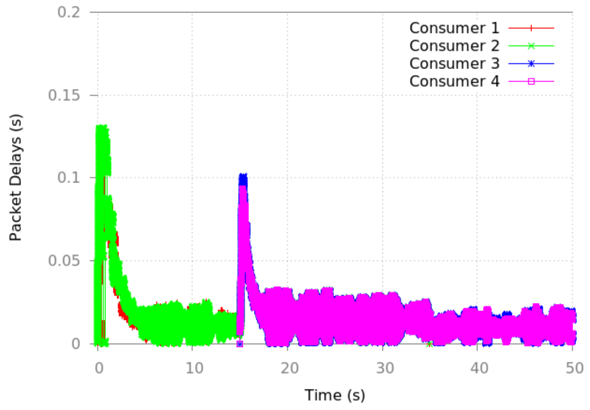
(c) HIS

Figure 5.34. Throughput

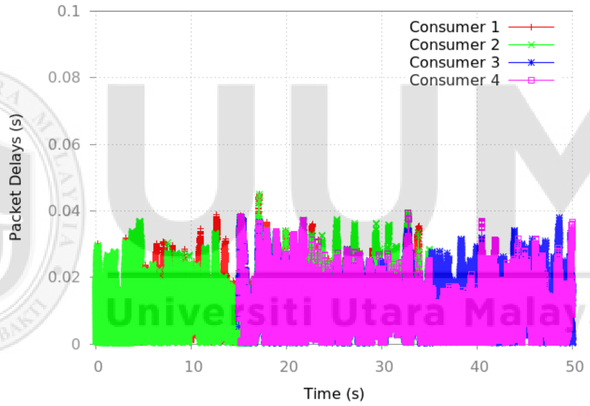
tion to consumers with different packet sizes and different starting times. In Figure 5.34 (b), showing the performance of PCON in the same scenario, adaptability starts at 25 sec of simulation time, which is 10 sec before consumers 1 and 2 stop. The adaptation is not fair among the four consumers, with instability between 2500 and 9000 packets. The bandwidth allocation is restored to 12000 for the remaining 15 sec for consumers 2 and 4. The performance of HIS is presented in Figure 5.34 (c) under the same settings as HRCM and PCON. The adaptability is seen at 25 sec, the same as for PCON, although consumers 1 and 2 were within 10 sec of their stopping time. The performance of HRCM against PCON and HIS is 71 % and 82%, respectively.

The delay is also observed with different starting times for consumers 1 and 2 and consumers 3 and 4, as described above. The delays for HRCM and PCON have some similarities, accounted for by the different start times allocated to consumers, both fluctuating around 0.02 except that HRCM jumps to 0.15 before stabilizing, as shown in Figure 5.35 (a) and 5.35 (b). The HIS delay fluctuation is around 0.05, as shown in Figure 5.35 (c), while for HRCM the delay decreased from 35 sec into simulation time until the end. This occurred with the ending of consumers 1 and 2, releasing extra bandwidth. Similarly, in HIS, stability started at 35 sec to 40 sec of simulation time. HRCM and HIS have common functionality, which is the sign of adaptability as presented in Figure 5.34 (a) and (c).

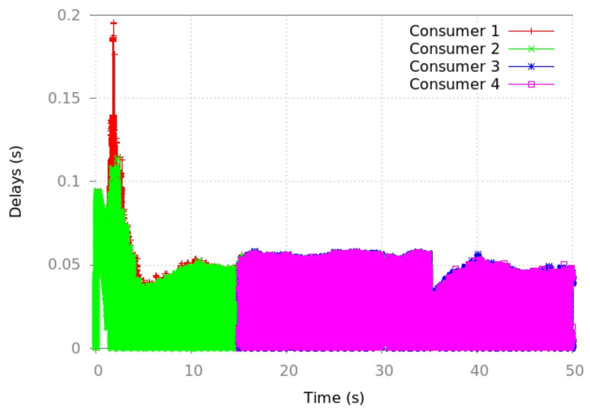
Figure 5.36 shows the link utilization and Figure 5.37 shows the fairness results under the same conditions. The link utilization result shows a sudden declination of the rate for all three mechanisms at 35 sec to 37 sec of simulation time. HRCM and PCON have similar patterns, slightly better for HRCM, while the HIS still shows its typical instability as described in the other results. Figure 5.37 shows the fairness and stability of link utilization for the three mechanisms. HRCM and HIS are very stable initially,



(a) HRCM



(b) PCON



(c) HIS

Figure 5.35. Delay

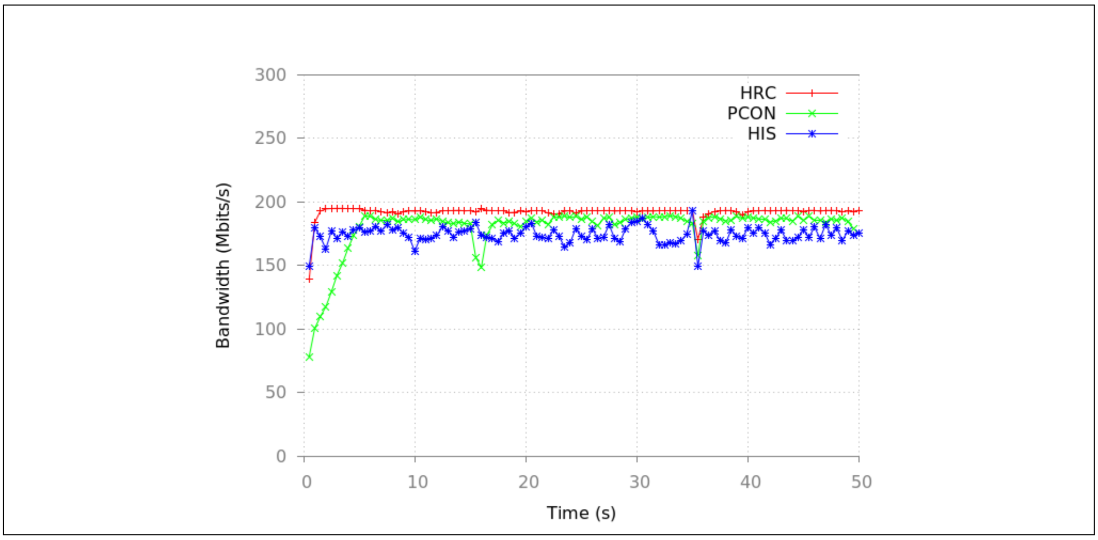


Figure 5.36. Link Utilization

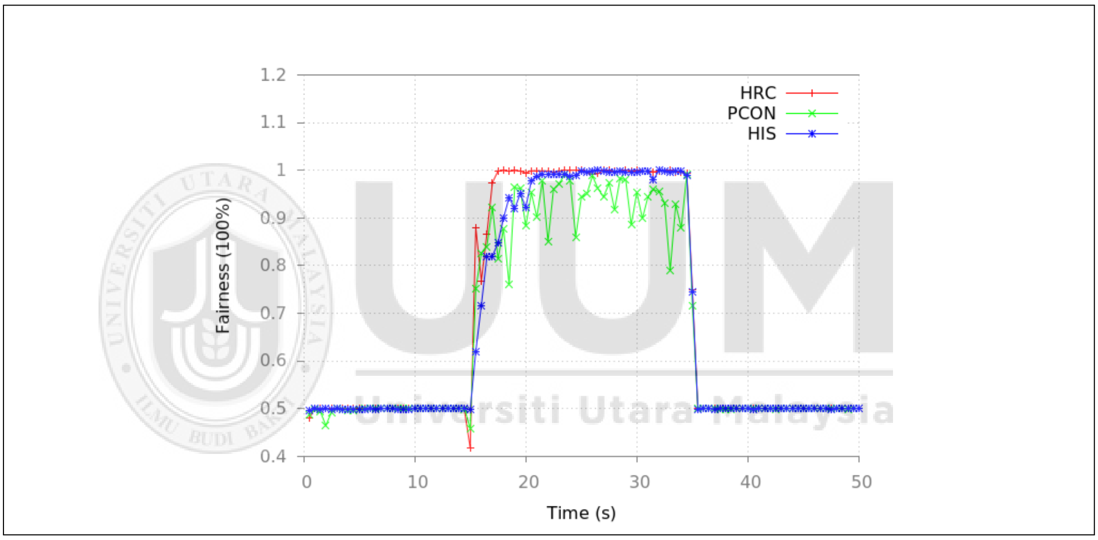


Figure 5.37. Fairness

while PCON keeps on fluctuating. At 15 sec simulation time, when the second pair of consumers start accessing the link, the rate changes in all mechanisms. HRCM increases to 100%, suddenly, while the rate of HIS inclines and reaches stability at 23 sec, where it coincides with HRCM. PCON fluctuates within the range without specific or fair action. In general, HRCM adapt to any changes as quickly as possible, while HIS takes around 8 sec slowly to reach the adaption level and PCON has no adaptability.

5.3 Summary

The HRCM schemes integration was discussed also its evaluation in ndnSIM simulation environment. It was evaluated by observing different performance metrics over a variety of scenarios. Specifically, it illustrates the effects of individual performance metrics, namely the Data packet size, multiple paths, and the different start time, , as well as a combination of these factors. The evaluation was accomplished by comparing the throughput, packet delay, queue length, download time, outgoing link utilization and fairness of HRCM, PCON and HIS in different scenarios. The findings demonstrate that HRCM improves window stability and therefore the forwarding rate, and maintains fairness between consumers in all topologies and scenarios. As the last section shows, HRCM improvement against HIS and PCON in terms of throughput is 75%, delay 20%, queue length 55%, link utilization 41%, fairness 20% and download time 20%. Overall, HRCM shows better results than the other schemes and is more suitable for Named Data Networks.

CHAPTER SIX
CONCLUSION AND FUTURE WORK

At the time when the Internet was designed, it played an important part in people's lives, running on top of the protocol stack of the Transmission Control Protocol/Internet Protocol (TCP/IP) to connect only a few machines under host-centric architecture. This host-driven Internet was superbly coordinated for early Internet use, which was not complicated. As, The initial Internet design was a model that shared little memory or resources when compared to the needs of present-day users. The growth in Internet use requiring improved usability resulted in a staggering increase in traffic and memory capacity to handle the data generated through user connectivity and host-to-host interactivity. This, along with the predicted exponential increase in traffic, gave rise to the leading research in Named Data Networking (NDN). The NDN architecture aims to bring information much closer to the subscribers by dissociating the address of the host and instead using content names. NDN has been agreed as a future paradigm of the in-network form of communication.

This thesis addressed the congestion issues of the transporting system and how this issue can affect the performance of forwarding in NDN. The previous chapters proposed a new mechanism, namely the Hybrid Rate Control Mechanism (HRCM) that consists of the Shaping Deficit Weight Round Robin (SDWRR), Queue-delay Parallel Multipath (QPM), and Agile-based Conservative Window Adaptation (EC-Agile) schemes. HRCM was implemented in ndnSIM and performance analysis was based on the numeric results obtained from the simulation.

This chapter summarizes the thesis in Section 6.1. Section 6.2 presents the research limitations, and Section 6.3 the contributions of this research. Finally, Section 6.4 makes recommendations for future study.

6.1 Summary of the Study

In this study, we introduced a Hybrid Rate Control Mechanism that jointly controls congestion from the end-users and within the network by in-path routers. We have shown that the receiver plays an important role in the rate control loop to guarantee full bandwidth link utilization and flow fairness. Hop-by-hop Interest shaping enhances rate and congestion control performance, and it is particularly suited to NDN for various reasons. Controlling Interest instead of Data packets gives the opportunity to prevent congestion by delaying Interest forwarding. Interest packets are smaller in size than Data packets, hence requiring smaller buffer capacity, and with reasonable buffers, dimensionless Interest losses may be avoided. Early congestion detection allows realizing early congestion control by locally monitoring per-prefix Interest queues at each uplink node and the corresponding Data queue on the downlink. An above-sojourn time threshold in the Interest or Data queue signals the beginning of congestion, before the detection of packet losses at the receiver. With protection from misbehaving receivers by shaping Interest in a hop-by-hop fashion, the greedy behaviour of a non-conformant receiver can be quickly detected and controlled in order to protect concurrent flows. HRCM reacts to misbehaving receivers by queuing Interest packets up to a certain threshold, before discarding them. Compared to the alternative solutions for NDN congestion control listed in Chapter Two, HRCM brings additional benefits through the SDWRR scheme's ability to monitor, control and forward the incoming packets as well as indicate congestion to inform consumers to reduce their rate. By this monitoring, coupled with NACK and Data marking, HRCM achieves optimal explicit feedback control. The available bandwidth is utilized without the need for external parameters like RTT, forwarding the Interest packets in parallel using QPM that adapts sojourn time as an internal parameter.

As set out in Chapter One, the aim of this research was to design a new hybrid rate-control mechanism that distributes the load within a network domain in a continuous manner and offloads congested links and paths. This control mechanism would have the ability to improve the forwarding mechanism by minimizing the link and caching overheads because it forwards the Interest packets in multipath mode, avoiding congestion. Moreover, this plane considers fairness among different types of flow. Chapter One also explained the main objectives and the significance of this work. A brief overview of the NDN architecture was provided as a future Internet paradigm. Next, the congestion problem was presented, showing that forwarding is affected and leading to increased delay, retransmission and packet drop that affects link utilization, fairness and the overall network performance. It was concluded that the current mechanism could not handle the challenge caused by content caching and size.

In addition, the unpredictable aggregation of Interest packets within the network and the important variation in RTT measurements, as a result of the in-network caching feature, prevent achieving fairness and handling the dynamics in returning data. Without a valuable congestion control scheme to guarantee effective fairness to assist the network resource, NDN cannot be considered as a complete working framework. In this case, fairness is an important issue in designing the NDN forwarding control strategies. Especially in the presence of competing flows, different packet sizes and misbehaving receivers, it is necessary to guarantee fair bandwidth allocation at the bottleneck link. The study also detailed the technical background of the research by reviewing the core properties of the NDN architecture that are essential in describing this work. The principles and operations of the congestion control and forwarding strategies were discussed in detail in Chapter Two.

In Chapter Three, a specific framework was introduced as guidelines to accomplish this research. In order to achieve the objectives, several methods were used. The researcher chose to focus on congestion control, as to answer all the questions posed by NDN would be totally outside the scope of the study.

Chapter Four elaborates the design of the proposed components of the HRCM. First, the SDWRR scheme shapes the Interest flow rate and indicates congestion at each interface; the basis of its design is queuing and scheduling theory. Eclipse was used for verification and simulation to validate the scheme, indicating that it improved the stability and fairness of the transport control. Second, QPM forwards incoming Interest packets to the available paths in FIB to utilize all available link bandwidth in each interface. Again, Eclipse was used for verification and simulation to validate the scheme and show that QPM is not affected by RTT. EC-Agile is the third contribution, designed to increase the forwarding rate in the incoming Data packets and decrease it on receiving a congestion notification or time-out. By using the two different feedback packets, NACK and packet marking, EC-Agile smoothly regulates the sending rate and is faster at adjusting than schemes after verification and validation.

The combination of SDWRR, QPM and EC-Agile in HRCM is a distinctive aspect of this research. Chapter Five presents an evaluation of the performance of HRCM as a whole, achieved by comparing the results obtained from the simulation experiments with those from PCON and HIS. The results demonstrated that HRCM improves throughput, fairness, delay, download time, and link utilization compared with the PCON and HIS. The findings have significant implications represented by providing a reliable mechanism to NDN architecture. The results of the HRCM performance evaluation emphasize the fact that the framed objectives of this research have been completely achieved.

6.2 Research Limitation

Although this research was conducted under careful selection and a methodical procedure, including the conceptual model, implementation, verification, validation and evaluation, it does have some limitations. Chiefly, this work was tested on specific network topologies widely used in such implementations and approved for NDN. However, it did not include all the previous mechanisms and different topologies because of limited time and resources. The focus is on transport control in router queues and FIB transport control rather than on other aspects of NDN architecture like routing protocols, PIT and CS.

6.3 Thesis Contribution

The overall contribution of this research was to design and implement a Shaping Deficit Weight Round Robin (SDWRR) scheme, Queue-delay Parallel Multipath (QPM) scheme and Explicit Congestion Agile-based Conservative Window Adaptation (EC-Agile) scheme and integrate them as the Hybrid Rate Control Mechanism (HRCM) in order to enhance link utilization, fairness and add stability to NDN transport control. The specific contributions of the research are as follows:

a. The design of the SDWRR congestion control scheme that hybridizes the queue delay and scheduling concepts to improve accuracy in avoiding congestion and enhancing fairness in NDN routers.

b. The design of QPM that uses SDWRR parameters to improve forwarding adaptation, link utilization and enhance the stability of the network.

c. The development of EC-Agile that reacts smoothly to SDWRR notification packets to improve consumer rate adaptation and enhance the Interest packet drop and re-transmission.

d. The merging of SDWRR, QPM and EC-Agile in HRCM improves the stability of the network by increasing throughput and link utilization and fairness. It achieves the proposed objectives, to distribute the load within a network domain in a continuous manner and offload congested links and paths.

e. The research contributes to the understanding of transport control in NDN, covering details of implementation, and weaknesses. Chapter Two covered the different concepts of congestion control proposed in earlier studies, and identified the lack of an appropriate mechanism. The new congestion mechanism fills these gaps.

6.4 Future Works

Finally, some topics can be suggested for future work:

a. The SDWRR model was designed using Fair Queue (FQ) queue scheduling; future work could implement other types of queue scheduling such as priority queuing (PQ) and weight fair queuing (WFQ), and evaluate them to identify the best performance.

b. Performance evaluation of the HRCM in a testbed would be another way to extend this research. Although HRCM was evaluated comprehensively and extensively through a validated simulator, its implementation in a real testbed is definitely of great interest. However, as the NDN architecture is not yet deployed, it would be complicated to evaluate its performance on the testbed. Nevertheless, evaluating HRCM using real traffic would undoubtedly be an excellent way of extending the scope of this research.

c. Because of the time and resource limits, this work was conducted in specific network topologies, whereas the real NDN is unpredictable and changeable. The researcher might consider using more complicated topologies to check the network performance.

d. One more suggestion for future research is to extend the work into the mobile and wireless environment, given the revolution in the use of mobile and wireless devices like smartphones, tablets and sensors. New application technology like IoT and V2V will concentrate the interest of researchers of the future Internet.

REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 1–12.

[2] S. Zhang, F. R. Yu, and V. C. Leung, "Joint connection admission control and routing in IEEE 802.16-based mesh networks," *IEEE transactions on wireless communications*, vol. 9, no. 4, pp. 1370–1379, 2010.

[3] Y. Wang, B. Xu, D. Tai, J. Lu, T. Zhang, H. Dai, B. Zhang, and B. Liu, "Fast name lookup for named data networking," in *IEEE 22nd International Symposium of Quality of Service (IWQoS)*. IEEE, 2014, pp. 198–207.

[4] A. K. M. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "NLSR: Named-data Link State Routing Protocol," in *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking*, ser. ICN '13. New York, NY, USA: ACM, 2013, pp. 15–20. [Online]. Available: http://doi.acm.org/10.1145/2491224.2491231

[5] J. Torres, L. Ferraz, and O. Duarte, "Controller-based routing scheme for named data network," *Electrical Engineering Program, COPPE/UFRJ, Tech: Rep*, 2012.

[6] H. Yuan, T. Song, and P. Crowley, "Scalable NDN forwarding: Concepts, issues and principles," in *21st International Conference on computer communications and networks (ICCCN)*. IEEE, 2012, pp. 1–9.

[7] I. Moiseenko and L. Zhang, "Consumer-producer api for named data networking," in *Proceedings of the 1st ACM Conference on Information-Centric Networking*. ACM, 2014, pp. 177–178.

[8] G. Carofiglio, M. Gallo, and L. Muscariello, "ICP: Design and evaluation of an interest control protocol for content-centric networking," in *Proceedings IEEE INFOCOM Workshops*. IEEE, 2012, pp. 304–309.

[9] H. Wu and H. Ma, "An optimal buffer management strategy for video transmission in opportunistic networks." *Adhoc & Sensor Wireless Networks*, vol. 29, 2015.

[10] Y. Ren, J. Li, S. Shi, L. Li, G. Wang, and B. Zhang, "Congestion control in named data networking–a survey," *Computer Communications*, vol. 86, pp. 1–11, 2016.

[11] T. Stockhammer, "Dynamic adaptive streaming over HTTP–: standards and design principles," in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 133–144.

[12] Q. Chen, R. Xie, F. R. Yu, J. Liu, T. Huang, and Y. Liu, "Transport control strategies in named data networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2052–2083, 2016.

[13] F. Zhang, Y. Zhang, A. Reznik, H. Liu, C. Qian, and C. Xu, "A Transport Protocol for Content-Centric Networking With Explicit Congestion Control," in *23rd International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2014, pp. 1–8.

[14] C. V. N. Index, "Forecast and methodology, 2013–2018," *Cisco white paper*, p. 154, 2014.

[15] B. L. Haibo, L. K. Sohraby, and C. Wang, "Future internet services and applications," *IEEE Network*, vol. 24, no. 4, pp. 4–5, 2010.

[16] A. Passarella, "A survey on content-centric technologies for the current Internet: CDN and P2P solutions," *Computer Communications*, vol. 35, no. 1, pp. 1–32, 2012.

[17] B. Frank, I. Poese, Y. Lin, G. Smaragdakis, A. Feldmann, B. Maggs, J. Rake, S. Uhlig, and R. Weber, "Pushing CDN-ISP collaboration to the limit," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 3, pp. 34–44, 2013.

[18] S. Kalarani and G. Uma, "Improving the efficiency of retrieved result through transparent proxy cache server," in *Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. IEEE, 2013, pp. 1–8.

[19] M. Handley, "Why the internet only just works," *BT Technology Journal*, vol. 24, no. 3, pp. 119–129, 2006.

[20] J. Rexford and C. Dovrolis, "Future internet architecture: clean-slate versus evolutionary research," *Communications of the ACM*, vol. 53, no. 9, pp. 36–40, 2010.

[21] A. Feldmann, "Internet clean-slate design: what and why?" *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 3, pp. 59–64, 2007.

[22] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A Survey of Information-Centric Networking Research," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 2, pp. 1024–1049, 2014.

[23] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A Data-oriented (and Beyond) Network Architecture," *SIGCOMM*, vol. 37, no. 4, pp. 181–192, Aug. 2007. [Online]. Available: http://doi.acm.org/10.1145/1282427.1282402

[24] V. Dimitrov and V. Koptchev, "PSIRP Project – Publish-subscribe Internet Routing Paradigm: New Ideas for Future Internet," in *Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference*

on Computer Systems and Technologies, ser. CompSysTech '10. New York, NY, USA: ACM, 2010, pp. 167–171. [Online]. Available: http://doi.acm.org/10.1145/1839379.1839409

[25] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of information (NETinf)–an information-centric networking architecture," *Computer Communications*, vol. 36, no. 7, pp. 721–735, 2013.

[26] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named Data Networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[27] J. Thompson and J. Burke, "NDN common client libraries," *Technical Report NDN-0024, Revision 1. NDN Project*, 2014.

[28] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in named data networking," *ACM SIGCOMM computer communication review*, vol. 42, no. 3, pp. 62–67, 2012.

[29] A. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "NLSR: named-data link state routing protocol," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*. ACM, 2013, pp. 15–20.

[30] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A case for stateful forwarding plane," *Computer Communications*, vol. 36, no. 7, pp. 779–791, 2013.

[31] S. Salsano, A. Detti, M. Cancellieri, M. Pomposini, and N. Blefari-Melazzi, "Transport-layer issues in information centric networks," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*. ACM, 2012, pp. 19–24.

[32] D. Byun, B.-J. Lee, and Y. Park, "Adaptive interest adjustment in CCN flow control," in *Global Communications Conference (GLOBECOM), 2014 IEEE*. IEEE, 2014, pp. 1873–1877.

[33] L. Saino, C. Cocora, and G. Pavlou, "CCTCP: A scalable receiver-driven congestion control protocol for content centric networking," in *IEEE international conference on communications (ICC)*. IEEE, 2013, pp. 3775–3780.

[34] A. Detti, C. Pisa, and N. Blefari-Melazzi, "Multipath forwarding strategies in Information Centric Networks with AIMD congestion control," *arXiv preprint arXiv:1412.2204*, 2014.

[35] T. Q. Nguyen, B.-P. Ha, T.-A. Le, P. L. Vo, T. T. Huynh, and C. H. Tran, "Coupled multipath congestion control at receiver in content-centric networking," in *2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*. IEEE, 2015, pp. 239–242.

[36] D. Nguyen, M. Fukushima, K. Sugiyama, and A. Tagami, "Efficient multipath forwarding and congestion control without route-labeling in CCN," in *IEEE international conference on communication workshop (ICCW)*. IEEE, 2015, pp. 1533–1538.

[37] J. Yao, B. Yin, and X. Tan, "A SMDP-based forwarding scheme in named data networking," *Neurocomputing*, vol. 306, pp. 213–225, 2018.

[38] N. Rozhnova and S. Fdida, "An effective hop-by-hop interest shaping mechanism for ccn communications," in *Proceedings IEEE INFOCOM Workshops*. IEEE, 2012, pp. 322–327.

[39] ——, "An extended hop-by-hop interest shaping mechanism for content-centric networking," in *Global Communications Conference (GLOBECOM), 2014 IEEE*. IEEE, 2014, pp. 1–7.

[40] G. Carofiglio, M. Gallo, and L. Muscariello, "Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*. ACM, 2012, pp. 37–42.

[41] Y. Wang, N. Rozhnova, A. Narayanan, D. Oran, and I. Rhee, "An improved hop-by-hop interest shaper for congestion control in named data networking," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 55–60.

[42] S. Mejri, H. Touati, N. Malouch, and F. Kamoun, "Hop-by-hop congestion control for named data networks," in *IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2017, pp. 114–119.

[43] T. KATO, M. BANDAI, and M. YAMAMOTO, "A congestion control method for named data networking with hop-by-hop window-based approach," *IEICE Transactions on Communications*, 2018.

[44] M. Amadeo, C. Campolo, and A. Molinaro, "Internet of things via named data networking: The support of push traffic," in *International Conference and Workshop on the Network of the Future (NOF)*. IEEE, 2014, pp. 1–5.

[45] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, and J. Cao, "Named data networking: a survey," *Computer Science Review*, vol. 19, pp. 15–55, 2016.

[46] Y. Ren, J. Li, S. Shi, L. Li, and G. Wang, "An explicit congestion control algorithm for named data networking," in *IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. IEEE, 2016, pp. 294–299.

[47] A. Kerrouche, M. R. Senouci, and A. Mellouk, "QoS-FS: A new forwarding strategy with QoS for routing in named data networking," in *IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–7.

[48] Y. Ye, B. Lee, R. Flynn, N. Murray, and Y. Qiao, "HLAF: Heterogeneous-Latency Adaptive Forwarding strategy for Peer-Assisted Video Streaming in NDN," in *IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2017, pp. 657–662.

[49] T. Kato and M. Bandai, "Congestion control avoiding excessive rate reduction in named data network," in *14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2017, pp. 108–113.

[50] Y. Liu, X. Piao, C. Hou, and K. Lei, "A cubic-based explicit congestion control mechanism in named data networking," in *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE, 2016, pp. 360–363.

[51] K. Schneider, C. Yi, B. Zhang, and L. Zhang, "A practical congestion control scheme for named data networking," in *Proceedings of the 2016 conference on 3rd ACM Conference on Information-Centric Networking*. ACM, 2016, pp. 21–30.

[52] A. Ndikumana, S. Ullah, K. Thar, N. H. Tran, B. J. Park, and C. S. Hong, "Novel cooperative and fully-distributed congestion control mechanism for content centric networking," *IEEE Access*, vol. 5, pp. 27 691–27 706, 2017.

[53] M. Wang, M. Yue, and Z. Wu, "WinCM: A Window based Congestion Control Mechanism for NDN," in *1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*. IEEE, 2018, pp. 80–86.

[54] K. M. Schneider and U. R. Krieger, "Beyond network selection: Exploiting access network heterogeneity with named data networking," in *Proceedings of the 2nd International Conference on Information-Centric Networking*. ACM, 2015, pp. 137–146.

[55] M. Wang, M. Yue, and Z. Wu, "WinCM: A Window based Congestion Control Mechanism for NDN," *IEEE International Conference on Hot Information-Centric Networking (HotICN 2018)*, 2018.

[56] D. W. M. Khademi, Naeem; Ros, "The New AQM Kids on the Block: Much Ado About Nothing?" Universitetet i Oslo, Tech. Rep., 2013. [Online]. Available: https://www.duo.uio.no/bitstream/handle/10852/37381/khademi-AQM_Kids_TR434.pdf

[57] T. Hoeiland-Joergensen, P. McKenney, D. Taht, J. Gettys, and E. Dumazet, "Flowqueue-codel," *IETF ID draft-ietf-aqm-fq-codel-00*, 2014.

[58] R. Adams, "Active Queue Management: A Survey," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 3, pp. 1425–1476, Third 2013.

[59] G. Carofiglio, M. Gallo, L. Muscariello, and M. Papali, "Multipath congestion control in content-centric networks," in *IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. IEEE, 2013, pp. 363–368.

[60] A. Udugama, X. Zhang, K. Kuladinithi, and C. Goerg, "An on-demand multipath interest forwarding strategy for content retrievals in CCN," in *IEEE network operations and management symposium (NOMS)*. IEEE, 2014, pp. 1–6.

[61] Y. Xin, Y. Li, W. Wang, W. Li, and X. Chen, "Content aware multi-path forwarding strategy in information centric networking," in *IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 2016, pp. 816–823.

[62] A. Detti, C. Pisa, and N. B. Melazzi, "Modeling multipath forwarding strategies in information centric networks," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2015, pp. 324–329.

[63] G. Carofiglio, M. Gallo, and L. Muscariello, "Optimal multipath congestion control and request forwarding in information-centric networks: Protocol design and experimentation," *Computer Networks*, vol. 110, pp. 104–117, 2016.

[64] A. Bouacherine, M. R. Senouci, and B. Merabti, "Parallel multi-path forwarding strategy for named data networking," in *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications*, vol. 1, 2016, pp. 36–46.

[65] C. Li, T. Huang, R. Xie, H. Zhang, J. Liu, and Y. Liu, "A novel multi-path traffic control mechanism in named data networking," in *22nd International Conference on Telecommunications (ICT)*. IEEE, 2015, pp. 60–66.

[66] K. M. Schneider, K. Mast, and U. R. Krieger, "CCN Forwarding Strategies for Multihomed Mobile Terminals," *arXiv preprint arXiv:1412.4422*, 2014.

[67] V. Jacobson, J. Burke, D. Estrin, L. Zhang, B. Zhang, G. Tsudik, K. Claffy, D. Krioukov, D. Massey, C. Papadopoulos *et al.*, "Named Data Networking (NDN) Project 2012-2013 Annual Report," Named Data Networking (NDN), Tech. Rep., 2013.

[68] D. Coilein and D. O'Mahony, "SAVANT: Aggregated feedback and accountability framework for named data networking," in *Proceedings of the 1st ACM Conference on Information-Centric Networking*. ACM, 2014, pp. 187–188.

[69] X. Jiang and J. Bi, "Interest set mechanism to improve the transport of named data networking," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 515–516.

[70] A. Compagno, M. Conti, P. Gasti, L. V. Mancini, and G. Tsudik, "Violating consumer anonymity: Geo-locating nodes in named data networking," in *International Conference on Applied Cryptography and Network Security*. Springer, 2015, pp. 243–262.

[71] X. Zeng and Z. H. Gao, "Rank-based routing strategy for named data network," in *Applied Mechanics and Materials*, vol. 543. Trans Tech Publ, 2014, pp. 3320–3323.

[72] M. Tortelli, L. A. Grieco, and G. Boggia, "Performance assessment of routing strategies in named data networking," in *IEEE ICNP*, 2013.

[73] M. Amadeo, C. Campolo, and A. Molinaro, "Forwarding strategies in named data wireless ad hoc networks: Design and evaluation," *Journal of Network and Computer Applications*, vol. 50, pp. 148–158, 2015.

[74] L. Wang, A. Afanasyev, R. Kuntz, R. Vuyyuru, R. Wakikawa, and L. Zhang, "Rapid traffic information dissemination using named data," in *Proceedings of the 1st ACM workshop on emerging name-oriented mobile networking design-architecture, algorithms, and applications*. ACM, 2012, pp. 7–12.

[75] M. Varvello, I. Rimac, U. Lee, L. Greenwald, and V. Hilt, "On the design of content-centric MANETs," in *Eighth International Conference on Wireless On-Demand Network Systems and Services*. IEEE, 2011, pp. 1–8.

[76] Y.-T. Yu, R. B. Dilmaghani, S. Calo, M. Sanadidi, and M. Gerla, "Interest propagation in named data manets," in *International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2013, pp. 1118–1122.

[77] M. Meisel, V. Pappas, and L. Zhang, "Ad hoc networking via named data," in *Proceedings of the fifth ACM international workshop on Mobility in the evolving internet architecture*. ACM, 2010, pp. 3–8.

[78] D. Saucez, L. A. Grieco, and C. Barakat, "AIMD and CCN: past and novel acronyms working together in the Future Internet," in *Proceedings of the 2012 ACM workshop on Capacity sharing*. ACM, 2012, pp. 21–26.

[79] C. Lim, Y. Kim, and I. Yeom, "Performance-based congestion control in information centric network," in *International Conference on Information Networking (ICOIN)*. IEEE, 2017, pp. 633–635.

[80] S. Braun, M. Monti, M. Sifalakis, and C. Tschudin, "An empirical study of receiver-based aimd flow-control strategies for CCN," in *22nd international conference on computer communication and networks (ICCCN)*. IEEE, 2013, pp. 1–8.

[81] K. Lei, J. Wang, and J. Yuan, "An entropy-based probabilistic forwarding strategy in named data networking," in *IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 5665–5671.

[82] Y. Ye, B. Lee, R. Flynn, N. Murray, G. Fang, J. Cao, and Y. Qiao, "PTP: Path-specified transport protocol for concurrent multipath transmission in named data networks," *Computer Networks*, vol. 144, pp. 280–296, 2018.

[83] S. Lee, Y. Kim, Y. Kim, and I. Yeom, "Active interest management for improving flow completion time in named-data networking," in *Proceedings of the 2014 CoNEXT on Student Workshop*. ACM, 2014, pp. 46–48.

[84] S. Lee, Y. Kim, I. Yeom, and Y. Kim, "Active request management in stateful forwarding networks," *Journal of Network and Computer Applications*, 2017.

[85] K. Lei, C. Hou, L. Li, and K. Xu, "A rcp-based congestion control protocol in named data networking," in *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*. IEEE, 2015, pp. 538–541.

[86] M. Mahdian, S. Arianfar, J. Gibson, and D. Oran, "MIRCC: Multipath-aware ICN Rate-based Congestion Control," in *Proceedings of the 2016 conference on 3rd ACM Conference on Information-Centric Networking*. ACM, 2016, pp. 1–10.

[87] Y. Ren, J. Li, S. Shi, L. Li, and X. Chang, "An interest control protocol for named data networking based on explicit feedback," in *Proceedings of the Eleventh ACM/IEEE Symposium on Architectures for networking and communications systems*. IEEE Computer Society, 2015, pp. 199–200.

[88] D. Posch, B. Rainer, and H. Hellwagner, "SAF: Stochastic Adaptive Forwarding in Named Data Networking," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 1089–1102, 2017.

[89] T. Fu, Y. Li, T. Lin, H. Tan, H. Tang, and S. Ci, "An effective congestion control scheme in content-centric networking," in *13th International Conference on Parallel and Distributed Computing, Applications and Technologies*. IEEE, 2012, pp. 245–248.

[90] A. J. Abu, B. Bensaou, and A. M. Abdelmoniem, "Inferring and Controlling Congestion in CCN via the Pending Interest Table Occupancy," in *IEEE 41st Conference on Local Computer Networks (LCN)*. IEEE, 2016, pp. 433–441.

[91] Q. Li, Y. Jiang, Y. Tan, and M. Xu, "Improving the transmission control efficiency in content centric networks," *Computer Communications*, 2017.

[92] J. Zhou, Q. Wu, Z. Li, M. A. Kaafar, and G. Xie, "A proactive transport mechanism with explicit congestion notification for NDN," in *IEEE international conference on communications (ICC)*. IEEE, 2015, pp. 5242–5247.

[93] S. Zhong, Y. Liu, J. Li, and K. Lei, "A rate-based multipath-aware congestion control mechanism in named data networking," in *IEEE international symposium on parallel and distributed processing with applications and 2017 IEEE international conference on ubiquitous computing and communications (ISPA/IUCC)*. IEEE, 2017, pp. 174–181.

[94] Q. Chen, R. Xie, F. R. Yu, J. Liu, T. Huang, and Y. Liu, "Transport control strategies in named data networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2052–2083, 2016.

[95] H. Park, H. Jang, and T. Kwon, "Popularity-based congestion control in named data networking," in *sixth international conference on ubiquitous and future networks (ICUFN)*. IEEE, 2014, pp. 166–171.

[96] J.-J. Kim, M.-W. Ryu, S.-H. Cha, and K.-H. Cho, "A traffic aware routing protocol for congestion avoidance in content-centric network," *International Journal of Multimedia & Ubiquitous Engineering*, vol. 9, no. 9, 2014.

[97] E. Yeh, T. Ho, Y. Cui, M. Burd, R. Liu, and D. Leong, "Vip: A framework for joint dynamic forwarding and caching in named data networks," in *Proceedings of the 1st international conference on Information-centric networking*. ACM, 2014, pp. 117–126.

[98] B. McCormick, F. Kelly, P. Plante, P. Gunning, and P. Ashwood-Smith, "Real time alpha-fairness based traffic engineering," in *Proceedings of the third workshop on Hot topics in software defined networking*. ACM, 2014, pp. 199–200.

[99] H. Qian, R. Ravindran, G.-Q. Wang, and D. Medhi, "Probability-based adaptive forwarding strategy in named data networking," in *IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. IEEE, 2013, pp. 1094–1101.

[100] P. Bazmi and M. Keshtgary, "A neural network based congestion control algorithm for content-centric networks," *Journal of Advanced Computer Science & Technology*, vol. 3, no. 2, p. 214, 2014.

[101] A. Karami, "ACCPndn: adaptive congestion control protocol in named data networking by learning capacities using optimized time-lagged feedforward neural network," *Journal of Network and Computer Applications*, vol. 56, pp. 1–18, 2015.

[102] L. Gong, J. Wang, X. Zhang, and K. Lei, "Intelligent forwarding strategy based on online machine learning in named data networking," in *IEEE Trustcom/BigDataSE/ISPA*. IEEE, 2016, pp. 1288–1294.

[103] Q. Huang and F. Luo, "Ant-colony optimization based qos routing in named data networking," *Journal of Computational Methods in Sciences and Engineering*, vol. 16, no. 3, pp. 671–682, 2016.

[104] A. Kerrouche, M. R. Senouci, A. Mellouk, and T. Abreu, "AC-QoS-FS: Ant Colony based QoS-aware Forwarding Strategy for Routing in Named Data Networking," in *IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[105] Y. Zhang, J. Wang, L. Gong, J. Yuan, and K. Lei, "A quantified forwarding strategy in ndn by integrating ant colony optimization into MADM," in *IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*. IEEE, 2017, pp. 214–220.

[106] J. Yao, B. Yin, X. Tan, and Y. Bao, "A game theoretic framework for congestion control in named data networking," *Information Technology And Control*, vol. 46, no. 4, pp. 605–618, 2017.

[107] O. Akinwande, "Interest forwarding in named data networking using reinforcement learning," *Sensors*, vol. 18, no. 10, p. 3354, 2018.

[108] J. Li, R.-c. Xie, T. Huang, and L. Sun, "A novel forwarding and routing mechanism design in SDN-based NDN architecture," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 9, pp. 1135–1150, 2018.

[109] T. Liu, M. Zhang, J. Zhu, R. Zheng, R. Liu, and Q. Wu, "ACCP: adaptive congestion control protocol in named data networking based on deep learning," *Neural Computing and Applications*, pp. 1–9, 2018.

[110] L. Chengming, L. Wenjing, and K. OKAMURA, "A greedy ant colony forwarding algorithm for named data networking," *Proceedings of the Asia-Pacific advanced network*, vol. 34, pp. 17–26, 2013.

[111] M. A. Alrshah, M. Othman, B. Ali, and Z. M. Hanapi, "Agile-SD: a Linux-based TCP congestion control algorithm for supporting high-speed and short-distance networks," *Journal of Network and Computer Applications*, vol. 55, pp. 181–190, 2015.

[112] R. Jain, *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons, 1990.

[113] V. V. Kalashnikov, *Mathematical methods in queuing theory*. Springer Science & Business Media, 2013, vol. 271.

[114] G. Thiruchelvi and J. Raja, "A survey on active queue management mechanisms," *International Journal of Computer Science and Network security*, vol. 8, no. 12, pp. 130–145, 2008.

[115] Y. Jiang, M. Hamdi, and J. Liu, "Self adjustable CHOKe: an active queue management algorithm for congestion control and fair bandwidth allocation," in *Proceedings of the Eighth IEEE Symposium on Computers and Communications. ISCC 2003*. IEEE, 2003, pp. 1018–1025.

[116] J. Koo, B. Song, K. Chung, H. Lee, and H. Kahng, "MRED: a new approach to random early detection," in *Proceedings 15th International Conference on Information Networking*. IEEE, 2001, pp. 347–352.

[117] S. Floyd, "TCP and explicit congestion notification," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 5, pp. 8–23, 1994.

[118] C. Long, B. Zhao, X. Guan, and J. Yang, "The yellow active queue management algorithm," *Computer Networks*, vol. 47, no. 4, pp. 525–550, 2005.

[119] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on networking*, no. 4, pp. 397–413, 1993.

[120] T. J. Ott, T. Lakshman, and L. H. Wong, "SRED: stabilized RED," in *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)*, vol. 3. IEEE, 1999, pp. 1346–1355.

[121] S. S. Kunniyur and R. Srikant, "An adaptive virtual queue (AVQ) algorithm for active queue management," *IEEE/ACM Transactions on Networking (ToN)*, vol. 12, no. 2, pp. 286–299, 2004.

[122] K. Nichols and V. Jacobson, "Controlling queue delay," *Queue*, vol. 10, no. 5, p. 20, 2012.

[123] A. Jaffri, Z. Ahmad, and M. Tahir, "Named data networking (NDN), new approach to future internet architecture design: a survey," *Int. J. Informatics Commun. Technol*, vol. 2, no. 3, pp. 155–164, 2013.

[124] T. Jain, B. Annappa, and M. P. Tahiliani, "Performance evaluation of CoDel for active queue management in wired-cum-wireless networks," in *Fourth International Conference on Advanced Computing & Communication Technologies*. IEEE, 2014, pp. 381–385.

[125] K. Katsaros, G. Xylomenos, and G. C. Polyzos, "MultiCache: An overlay architecture for information-centric networking," *Computer Networks*, vol. 55, no. 4, pp. 936–947, 2011.

[126] A. Detti, N. Blefari Melazzi, S. Salsano, and M. Pomposini, "CONET: a content centric inter-networking architecture," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*. ACM, 2011, pp. 50–55.

[127] R. Ouni, J. Bhar, and K. Torki, "A new scheduling protocol design based on deficit weighted round robin for qos support in ip networks," *Journal of Circuits, Systems and Computers*, vol. 22, no. 03, p. 1350012, 2013.

[128] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin," in *ACM SIGCOMM Computer Communication Review*, vol. 25, no. 4. ACM, 1995, pp. 231–242.

[129] L. Blessing and A. Chakrabarti, *DRM: A Design Research Methodology*. Springer Verlag, 2009.

[130] B. Moore, *Eclipse Development: Using the Graphical Editing Framework and the Eclipse Modeling Framework*. Books24x7.com, 2004.

[131] M. H. Omar, "An innovative signal detection algorithm in facilitating the cognitive radio functionality for wireless regional area network using singular value decomposition," phdthesis, Universiti Utara Malaysia, 2011.

[132] X. Chang, "Network simulations with OPNET," in *WSC'99. 1999 Winter Simulation Conference Proceedings.'Simulation-A Bridge to the Future'(Cat. No. 99CH37038)*, vol. 1. IEEE, 1999, pp. 307–314.

[133] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*. ICST (Institute for Computer Sciences, Social-Informatics and . . . , 2008, p. 60.

[134] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and tools for network simulation*. Springer, 2010, pp. 15–34.

[135] R. Chiocchetti, D. Rossi, and G. Rossini, "CCNsim: An highly scalable CCN simulator," in *IEEE International Conference on Communications (ICC)*. IEEE, 2013, pp. 2309–2314.

[136] M. Tortelli, D. Rossi, G. Boggia, and L. A. Grieco, "ICN software tools: survey and cross-comparison," *Simulation Modelling Practice and Theory*, vol. 63, pp. 23–46, 2016.

[137] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2.0: A new version of the NDN simulator for NS-3," *NDN, Technical Report NDN-0028*, 2015.

[138] J. Pan and R. Jain, "A survey of network simulation tools: Current status and future developments," *Email: jp10@ cse. wustl. edu*, vol. 2, no. 4, p. 45, 2008.

[139] K. Pentikousis, B. Ohlman, D. Corujo, G. Boggia, G. Tyson, E. Davies, A. Molinaro, and S. Eum, "Information-centric networking: baseline scenarios," Tech. Rep., 2015.

[140] CCNPL-Sim. Ccn packet level simulator (ccnpl-sim),. [Online]. Available: http://systemx.enst.fr/ccnpl-sim

[141] A. Afanasyev, I. Moiseenko, L. Zhang *et al.*, "ndnSIM: NDN simulator for NS-3," *University of California, Los Angeles, Tech. Rep*, vol. 4, 2012.

[142] C. Yi, "Adaptive forwarding in named data networking," phdthesis, Department of Computer Science, University of Arizona, 2014.

[143] N. Rozhnova, "Congestion control for content-centric networking," phdthesis, Universit´e Pierre et Marie Curie - Paris VI, 2015.

[144] Y. Wang, "Caching, routing and congestion control in a future information-centric internet," phdthesis, Raleigh, North Carolina, 2013.

[145] L. M. Mengue, "Cache, process and forward in information-centric networking," phdthesis, TELECOM ParisTech, 2016.

[146] Balci, O., "Verification Validation and Accreditation of Simulation Models," in *Proceedings of the 29th conference on Winter simulation*. IEEE Computer Society, 1997, pp. 135–141.

[147] R. G. Sargent, "Verification and Validation of Simulation Models," in *Proceedings of the 37th conference on Winter simulation*. Winter Simulation Conference, 2005, pp. 130–143.

[148] ——, "Verification and validation of simulation models," in *Proceedings of the 2010 winter simulation conference*. IEEE, 2010, pp. 166–183.

[149] Balci, Osman, "Validation, Verification, and Testing Techniques Throughout the Life Cycle of a Simulation Study," *Annals of operations research*, vol. 53, no. 1, pp. 121–173, 1994.

[150] O. Balci, "Golden rules of verification, validation, testing, and certification of modeling and simulation applications," *SCS M&S Magazine*, vol. 4, no. 4, pp. 1–7, 2010.

[151] R. B. Whitner and O. Balci, "Guidelines for selecting and using simulation model verification techniques," in *Proceedings of the 21st conference on Winter simulation*, 1989, pp. 559–568.

[152] L. G. Birta and F. N. Özmizrak, "A knowledge-based approach for the validation of simulation models: the foundation," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 6, no. 1, pp. 76–98, 1996.

[153] R. S. Pressman, *Software engineering: a practitioner's approach*. Palgrave macmillan, 2005.

[154] G. J. Myers, C. Sandler, and T. Badgett, *The art of software testing*. John Wiley & Sons, 2011.

[155] A. Habbal, "TCP SINTOK: Transmission Control Protocol with Delay-based Loss Detection and Contention Avoidance Mechanisms For Mobile Ad hoc Networks," phdthesis, School of Computing, Universiti Utara Malaysia, 2014.

[156] B. Beizer, *Software testing techniques*. Dreamtech Press, 2003.

[157] R. L. Van Horn, "Validation of simulation results," *Management Science*, vol. 17, no. 5, pp. 247–258, 1971.

[158] C. F. Hermann, "Validation problems in games and simulations with special reference to models of international politics," *Behavioral science*, vol. 12, no. 3, pp. 216–231, 1967.

[159] K. J. Cohen and R. M. Cyert, "Computer models in dynamic economics," *The Quarterly Journal of Economics*, vol. 75, no. 1, pp. 112–127, 1961.

[160] R. D. Wright, "Validating dynamic models: An evaluation of tests of predictive power," in *Proceedings of the 1972 Summer Computer Simulation Conference*. Simulation Councils, Inc La Jolla, California, 1972, pp. 14–16.

[161] S. Hassan and M. Kara, "Simulation-based performance comparison of TCP-friendly congestion control protocols," in *Proceedings of the 16th Annual UK Performance Engineering Workshop (UKPEW2000)*, 2000, pp. 199–210.

[162] A. Ahmad Suki Che Mohamed, "Slight-Delay Shaped Variable Bit Rate (SDSVBR) Technique for Video Transmission," phdthesis, College of Arts and Sciences, Universiti Utara Malaysia, 2011.

[163] G. Osman, "Scaleable and Smooth TCP-Friendly Receiver-Based Layered Multicast Protocol," phdthesis, Universiti Utara Malaysia, 2008.

[164] R. N. K. Alubady, "An efficient pending interest table control management in named data network," phdthesis, Universiti Utara Malaysia, 2017.

[165] A. N. Shahrudin, "Enhancing datagram congestion control protocol for efficient long delay link," phdthesis, Universiti Utara Malaysia, 2012.

[166] R. Jain, D.-M. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," 1998.

[167] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to end congestion avoidance on a global Internet," *IEEE Journal on selected Areas in communications*, vol. 13, no. 8, pp. 1465–1480, 1995.

[168] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Queue*, vol. 14, no. 5, p. 50, 2016.

[169] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4. IEEE, 2004, pp. 2514–2524.

[170] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-friendly High-Speed TCP Variant," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 5, pp. 64–74, 2008.